

UNIVERSIDAD CARLOS III DE MADRID  
TRABAJO FIN DE GRADO



**DISEÑO E IMPLEMENTACIÓN DE UNA ESTACIÓN METEOROLÓGICA  
PORTABLE PARA TELÉFONOS MÓVILES INTELIGENTES**

**Autor:** David Pedraza López

**Titulación:** Grado en Ingeniería Electrónica Industrial y Automática

**Profesor Tutor:** Braulio García Cámara (Departamento de Tecnología Electrónica)

**Codirector:** Josué Pagán Ortiz (Universidad Complutense de Madrid, Departamento de Arquitectura de Computadores y Automática)



**Título:** Diseño e implementación de una estación meteorológica portable para teléfonos móviles inteligentes.

**Autor:** David Pedraza López.

**Tutor:** Braulio García Cámara.

**Codirector:** Josué Pagán Ortiz.

**EL TRIBUNAL**

**Presidente:** Marta Ruiz Llata.

**Vocal:** Ricardo Valverde Gil.

**Secretario:** Alberto Emanuel Quintero Gámez.

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 4 de Julio de 2016 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

**VOCAL**

**SECRETARIO**

**PRESIDENTE**



## Agradecimientos.

Poco a poco se van tachando las fechas del calendario y se va poniendo fin a otra etapa, nunca está de más mirar atrás, observar todo lo que ha ocurrido todo este tiempo y darse cuenta lo mucho que debo agradecer, por lo que espero no olvidarme de nadie.

Lo primero agradecer a Braulio y Josué, mis tutores, esos envíos de correos bomba, algunos incluso a horas intempestivas o las reuniones por videoconferencia, os habéis volcado conmigo. Sin vosotros este proyecto no sería tal.

A mis padres Salvador e Isabel, por el esfuerzo que habéis tenido que hacer, siempre queriendo darme lo mejor. Y por hacerme ver las cosas de otra manera, cuando estoy a punto de desesperar.

A mi hermano Salva, el claro ejemplo de que aunque nuestras vidas lleven caminos muy distintos, un hermano siempre está ahí. Nunca olvidaré una conversación en la puerta de un hospital que me hicieron luchar por esta carrera.

A las nuevas amistades, a las de dentro de la universidad como Jonathan y Raquel. Dicen que el roce hace el cariño y nosotros hemos compartido muchas horas, clases y proyectos, solo digo eso. Y a las de fuera, Isma, tú que tienes una mente diferente al resto, tu presencia o tu ausencia siempre se hace de notar.

Y a las viejas amistades. Como Diego, con quien siempre se puede salir a dar un paseo y tener una larga conversación. O Ana, quien me ha ayudado a desarrollar la App de este proyecto.

También a los compañeros de Instituto que por suerte seguimos compartiendo momentos en la universidad. A C.Donderis, que me allanaste los primeros años de universidad. Tu positivismo, tu esfuerzo y capacidad de trabajo se contagia. También a Víctor, que seguro que tenías cosas mejores que hacer en Semana Santa que ayudarme en lo que pudiste.

Por último, no quería acabar este apartado sin acordarme de cuatro personas que por desgracia no leerán estas líneas. Que me habéis apoyado muchísimo, ya sea mostrándome lo bueno que puede estar un bocadillo de tiras, o la simple compañía en las tardes de estudio, llegando a acompañarme domingo tras domingo al campo de fútbol, dando igual el tiempo que haga o recogándome cada día del colegio durante años y quedándoos conmigo hasta la cena, no os olvidaré nunca.

De verdad y de todo corazón muchísimas gracias a todos.

"Me lo contaron y lo olvidé, lo vi y lo aprendí, lo hice y lo entendí." Confucio





## Resumen.

El objetivo de este proyecto es el diseño e implementación de una estación meteorológica portátil, para la integración en un estudio de afección de variables ambientales en la enfermedad de migraña.

La estación será capaz de medir temperatura, humedad relativa, presión atmosférica, nivel de presión sonora y luz ambiente. Se pretende que este sistema embebido se comunique con un dispositivo inteligente que ejecute una aplicación software de monitorización que colecta, almacena y envía los datos a un servidor.

Este proyecto se integra en un estudio de investigación que realiza el grupo de investigación GreenDisc (Universidad Complutense de Madrid) en colaboración con el Hospital Universitario de La Princesa sobre el estudio predictivo de la enfermedad de migraña. La migraña es una enfermedad neurológica incapacitante que genera un alto coste a los Servicios Nacionales de Salud en todo el mundo; además de la repercusión negativa que tiene para la calidad de vida de quien la padece y el problema social que supone. En el estudio se monitorizan variables biomédicas del paciente migrañoso, y se trata de establecer una relación entre esos datos y los ataques de migraña a través de modelos predictivos y técnicas de clasificación. Prediciendo una crisis de migraña se puede anticipar la toma del medicamento y evitar el dolor.

Numerosos estudios relacionan las crisis de migraña con cambios en las variables ambientales. Este proyecto tratará de aportar al estudio datos de variables ambientales que puedan afectar al inicio de una crisis de migraña. Para ello, se ha desarrollado un sistema que integra sensores para monitorizar las variables ambientales deseadas y un firmware que, ejecutado sobre un microcontrolador, gestione la estación meteorológica y el manejo de datos, con un dispositivo inteligente. El código del microcontrolador se encargará de reunir los datos de los sensores de temperatura, humedad, presión atmosférica, y nivel de presión sonora, y comunicarlos a una aplicación que correrá en un dispositivo inteligente Android mediante un protocolo de comunicación adecuado para la interfaz de audio por la que se comunican.

Por último, se ha trabajado tratando de alimentar la estación meteorológica desde el dispositivo móvil a través de la interfaz de audio de éste. Se ha estudiado la energía que es capaz de proporcionar el teléfono a través de dicha interfaz, y adaptada por un circuito de recolección de energía o *energy harvesting* para alimentar el sistema. Los resultados de estas pruebas no resultaron fructíferos, por lo que se optó por otras vías de alimentación. No obstante, al final de este trabajo se apuntan posibles líneas de futuras de actuación.





## Abstract.

The objective of this project is the design and implementation of a portable weather station for integration into a study of environmental variables on condition of migraine disease.

The station will be capable of measuring temperature, relative humidity, atmospheric pressure, sound pressure level and ambient light. It is intended that this embedded system will be able to communicate with an intelligent device that runs a software monitoring application that collects, stores and sends the data to a server.

This project is part of a research study conducted by the research group GreenDisc (Universidad Complutense de Madrid) in collaboration with the University Hospital of La Princesa on the predictive study of disease migraine. Migraine is a disabling neurological disease that generates a high cost to the National Health Services worldwide, in addition to the negative impact it has for the quality of life of the sufferer and the inherent social problem. Biomedical variables in the study of migraine patients are monitored, trying to establish a relationship between these data and migraine attacks through predictive models and classification techniques. Predicting a migraine can anticipate taking the drug and avoid pain.

Many studies have linked migraine attacks with changes in environmental variables. This project would seek to provide data to the study of environmental variables that can affect the onset of a migraine. To this end, it has developed a system that integrates sensors to monitor the desired environmental variables and a firmware, executed on a microcontroller, managing the weather station and data management, with an intelligent device. The microcontroller code is responsible for collecting data from sensors of temperature, humidity, atmospheric pressure and sound pressure level, and communicate to an application that will run on an Android smart device using a suitable communication protocol for audio interface by communicating.

Finally, we have worked trying to feed the weather station from the mobile device through this audio interface. We have studied the energy that is able to provide the phone via the interface, and a circuit adapted by energy harvesting to power the system. The results of these tests were not successful, so we opted for other supply options. However, at the end of this work possible future courses of action are targeted.



## Índice general.

<b>AGRADECIMIENTOS.</b>	<b>5</b>
<b>RESUMEN.</b>	<b>7</b>
<b>ABSTRACT.</b>	<b>9</b>
<b>ÍNDICE GENERAL.</b>	<b>11</b>
<b>ÍNDICE DE FIGURAS.</b>	<b>13</b>
<b>ÍNDICE DE TABLAS.</b>	<b>15</b>
<b>CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS.</b>	<b>17</b>
1.1. INTRODUCCIÓN.	17
1.1.1. <i>Motivación: La influencia de los parámetros ambientales en los enfermos de migraña.</i>	17
1.1.1.1. La migraña.	17
1.1.1.2. El proyecto de investigación.	19
1.1.2. <i>¿Existe algún dispositivo comercial?</i>	20
1.2. OBJETIVOS.	22
1.2.1. <i>Objetivo Principal.</i>	22
1.2.2. <i>Objetivos secundarios.</i>	22
1.3. ESPECIFICACIONES DEL SISTEMA.	23
1.4. FASES DEL PROYECTO.	24
1.5. MEDIOS UTILIZADOS.	25
1.5.1. <i>Instalaciones.</i>	25
1.5.2. <i>Hardware.</i>	25
1.5.3. <i>Herramientas de Software.</i>	26
<b>CAPÍTULO 2. DISEÑO DEL SISTEMA HARDWARE.</b>	<b>29</b>
2.1. DISEÑO DEL SISTEMA COMPLETO.	29
2.2. DISEÑO DEL SISTEMA HARDWARE.	30
2.2.1. <i>Requisitos y características detalladas de los sensores.</i>	30
2.2.2. <i>Selección.</i>	31
2.2.2.1. Temperatura.	35
2.2.2.2. Luz.	35
2.2.3. <i>Esquema general del sistema sensor.</i>	35
2.2.3.1. I <sup>2</sup> C.	36
2.2.3.2. Sensor de nivel de presión sonora.	37
2.2.3. <i>Implementación y pruebas iniciales.</i>	38
2.2.3.1. Pruebas hardware.	39
2.2.3.1.1. Sensor de presión sonora.	40
2.3. DISEÑO DEL SISTEMA DE ALIMENTACIÓN: <i>ENERGY HARVERSTER.</i>	42
2.3.1. <i>Obtención de energía del teléfono.</i>	42
2.3.2. <i>Diseño inicial del sistema de alimentación.</i>	44
2.3.3. <i>Implementación y pruebas iniciales.</i>	44
2.3.4. <i>Conclusiones y sistema final.</i>	47
2.4. ADAPTADOR DE IMPEDANCIAS.	48
<b>CAPÍTULO 3. DISEÑO DEL SISTEMA SOFTWARE.</b>	<b>53</b>
3.1. SENSOR DE LUMINOSIDAD.	53
3.2. SISTEMA DE COMUNICACIONES.	54
3.2.1. <i>Codificación/decodificación microcontrolador-Android.</i>	54
3.2.2. <i>Comunicación FSK.</i>	55
3.2.2.1. Firmware del microcontrolador.	56
3.2.2.2. Aplicación Android.	56
3.3. OTRAS FUNCIONES INCORPORADAS.	58
3.3.1. <i>Almacenamiento de datos.</i>	58
<b>CAPÍTULO 4. IMPLEMENTACIÓN FINAL Y PRUEBAS DE FUNCIONAMIENTO.</b>	<b>59</b>
4.1. IMPLEMENTACIÓN FINAL DEL SISTEMA ELECTRÓNICO.	59
4.2. IMPLEMENTACIÓN FINAL DEL SISTEMA SOFTWARE.	60
4.2.1. <i>Tiempo de envío.</i>	60
4.3 PRUEBAS DE FUNCIONAMIENTO.	61

4.3.1. Pruebas de la electrónica del sistema.....	61
4.3.2. Pruebas experimentales.....	62
4.3.2.1. Pruebas de comunicación.....	62
4.3.3. Situación real. ....	62
<b>CAPÍTULO 5. CONCLUSIONES Y POSIBLES LÍNEAS FUTURAS. ....</b>	<b>63</b>
5.1. CONCLUSIONES.....	63
5.2. LÍNEAS FUTURAS. ....	63
<b>BIBLIOGRAFÍA. ....</b>	<b>65</b>
<b>LISTADO DE ACRÓNIMOS.....</b>	<b>67</b>
<b>ANEXOS .....</b>	<b>69</b>
A.1 ESQUEMÁTICOS Y LAYOUTS DE LOS CIRCUITOS.....	71
A.1.1. Esquemático estación meteorológica.....	71
A.1.2. Layout circuito harvester capa TOP.....	73
A.1.3. Layout circuito harvester capa BOTTOM.....	75
A.1.4. Esquemático adaptador de impedancias original.....	77
A.1.5. Esquemático adaptador de impedancias adaptado.....	79
A.1.6. Esquemático circuito harvester. ....	81
A.1.7. Layout estación meteorológica capa TOP.....	83
A.1.8. Layout estación meteorológica capa BOTTOM.....	85
A.2 ENLACES A HOJAS DE CARACTERÍSTICAS.....	87
A.2.1. Datasheet LPR6235. ....	87
A.3. VIDEOS DE PRUEBAS. ....	89
A.3.1. Video alimentación placa de pruebas. ....	89
A.3.2. Video prueba sensor BMP180.....	89
A.3.3. Video calibración sensor de nivel de presión sonora. ....	89
A.3.4. Video prueba de comunicación con monitor serial. ....	89
A.3.5. Video prueba de comunicación del sistema completo. ....	89
A.3.6. Video prueba de consumo microcontrolador. ....	89
A.3.7. Video prueba dormir microcontrolador.....	90
A.3.8. Video prueba despertar microcontrolador desde el teléfono.....	90
A.4. OTRAS FUNCIONALIDADES HARDWARE. ....	91
A.4.1. Reducción de frecuencia ATmega328P. ....	91
A.4.2. Dormir microcontrolador y despertarlo desde el teléfono móvil.....	93
A.5. CÓDIGOS DE IDENTIFICACIÓN DE LAS VARIABLES. ....	95
A.6. PUERTOS DEL MICROCONTROLADOR. ....	97
A.7. PRUEBAS FUNCIONALES QUE NO SE PUEDEN INCORPORAR EN EL DISEÑO. ....	99
A.7.1. Reducción de frecuencia dinámica ATmega328.....	99
A.8. DISEÑO CAJA DE PROTECCIÓN. ....	101
A.9. PLIEGO DE CONDICIONES. ....	103
A.9.1. Pliegos de especificaciones generales. ....	103
A.9.2. Pliegos de especificaciones técnicas.....	103
A.9.2.1. Smartphone.....	103
A.9.2.2. Estación meteorológica.....	103
A.9.2.2.1. Conector jack.....	103
A.9.2.2.2. Sensores. ....	103
A.9.2.2.2.1. Sensor de humedad.....	103
A.9.2.2.2.2. Sensor de presión. ....	103
A.9.2.2.2.3. Sensor de temperatura.....	103
A.9.2.2.2.4. Sensor de luz.....	103
A.10. PRESUPUESTO. ....	105



## Índice de Figuras.

TIPOS DE CEFALEA.....	17
SÍNTOMAS PRODRÓMICOS .....	18
MANIFESTACIONES CLÍNICAS.....	18
APLICACIÓN MIGRAINE.....	20
INDICACIÓN GRADOS DE DOLOR.....	20
ACCESORIO MARCA SHAKA .....	21
WIDOO .....	21
DISPOSITIVOS VALIDOS .....	22
DIAGRAMA DE GANNT.....	24
PLACA ARDUINO UNO .....	25
LOGO ARDUINO .....	26
LOGO ANDROID STUDIO .....	26
LOGO EAGLE .....	26
LOGO GERBERLOGIX.....	26
LOGO AUDACITY .....	27
LOGO <i>SIGNAL GENERATOR</i> .....	27
LOGO SKETCHUP.....	27
DIAGRAMA DE FUNCIONAMIENTO DEL SISTEMA .....	29
MPL3115A2 .....	32
BMP 280 .....	32
MS563702BA03-50.....	32
MPX5100 .....	32
BMP 180 .....	32
CHS-MSS .....	33
HTU21D .....	33
SHT21.....	33
CHIPCAP 2.....	33
SPH0641LM4H1 .....	34
SPW2430HR5HB .....	34
SPH0642HT5H-1 .....	34
AKU143 .....	34
ESQUEMÁTICO DE SENSORES.....	36
COMUNICACIÓN I2C.....	37
ESQUEMÁTICO CONVERTIDOR .....	38
ESQUEMÁTICO CIRCUITO <i>HARVESTER</i> CON AMPLIFICADOR OPERACIONAL .....	39
<i>PRESALER</i> EN ATMEGA328 <sup>29</sup> .....	41
COMPARATIVA ADQUISICION DOS ONDAS .....	41
POTENCIA OBTENIDA EN MOTOROLA MOTO G.....	43
POTENCIA OBTENIDA BQ AQUARIS .....	44
POTENCIA <i>HARVESTER</i> .....	45
COMPARATIVA POTENCIAS ENTREGADAS .....	46
1. ONDA FORMATO <i>WAV</i> (IZQUIERDA). 2. ONDA FORMATO <i>MP3</i> (CENTRO). 3. ONDA FORMATO <i>MP3</i> (DERECHA) .....	47
CONECTOR JACK DE AUDIO TRRS.....	48
IMPLEMENTACIÓN PLACA DE PRUEBAS DEL ADAPTADOR DE IMPEDANCIA.....	49
SEÑAL EMITIDA POR EL MICROCONTROLADOR.....	49
COMPARATIVA SEÑALES.....	50
SEÑAL CON LAS RESISTENCIAS INVERTIDAS. ....	50
SEÑAL FINAL ADAPTADOR DE IMPEDANCIAS.....	51
INTERFAZ DE USUARIO DE APLICACIÓN DE LUZ.....	54
COMPARACIÓN DE SEÑALES FSK.....	56
INTERFAZ DE USUARIO ANDROIDTERM ADAPTADA.....	57
DIAGRAMA PROYECTO .....	59
FOTOGRAFÍA PLACA DE PRUEBAS.....	60

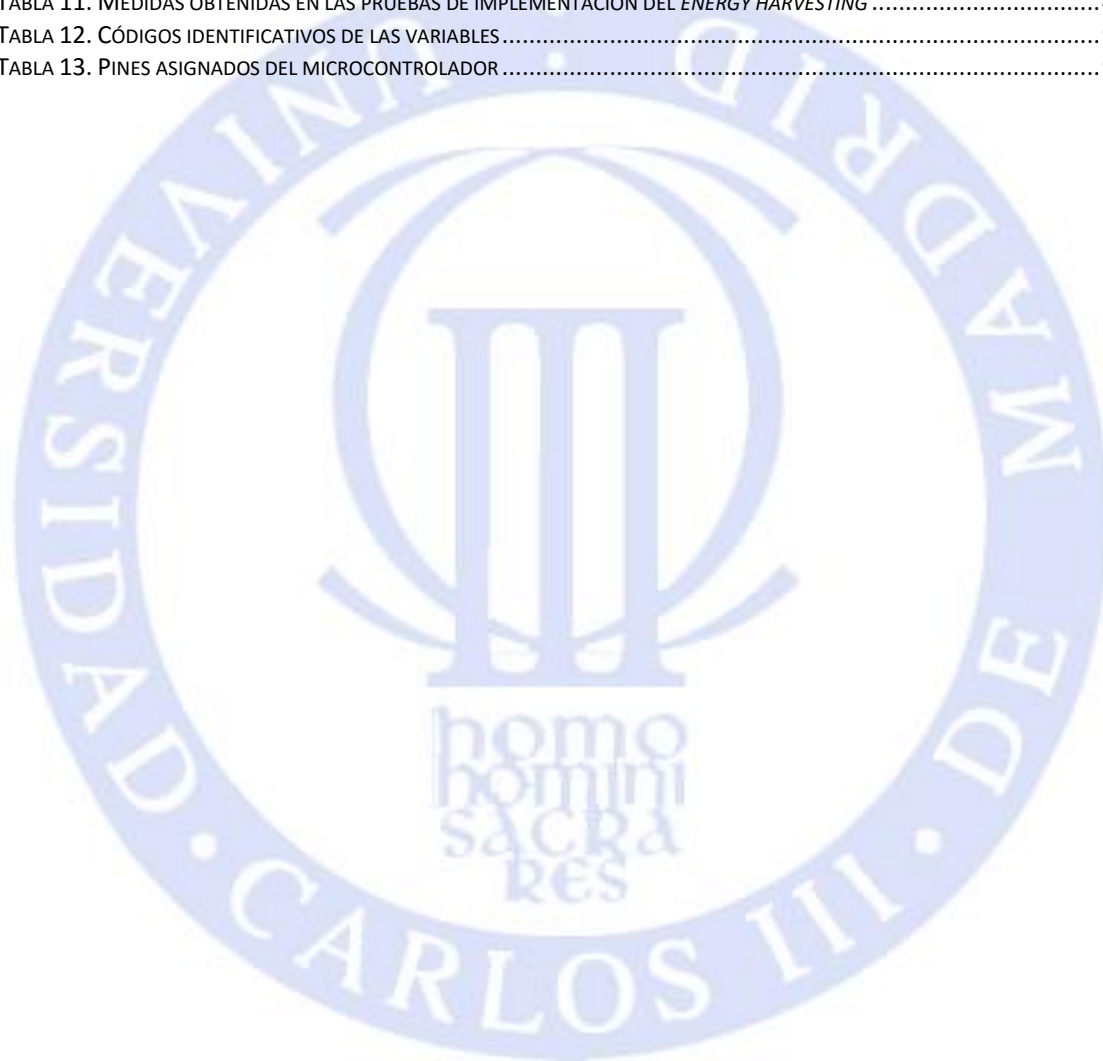
SISTEMA COMPLETO.....	60
PRUEBA DE MEDICIÓN DE SONIDO.....	61
COMPARATIVA DE LAS 5 VARIABLES.....	62
MICROCONTROLADOR EN PROTOBOARD.....	91
ARCHIVO BOOTLOADER.....	92
CONEXIONADO PARA CARGAR EL BOOTLOADER.....	92
GRÁFICA CONSUMO ATMEGA328P.....	93
BASE Y TAPA DE LA CARCASA.....	101
FOTOGRAFÍA CARCASA.....	101





## Índice de tablas.

TABLA 1. PUNTUACIONES DE LOS SENSORES.....	31
TABLA 2. CARACTERÍSTICAS SENSORES DE PRESIÓN.....	32
TABLA 3. PUNTUACIONES SENSOR DE PRESIÓN .....	32
TABLA 4. CARACTERÍSTICAS SENSORES DE HUMEDAD .....	33
TABLA 5. PUNTUACIONES SENSOR DE HUMEDAD .....	33
TABLA 6. CARACTERÍSTICAS SENSORES DE NIVEL DE PRESIÓN SONORA .....	34
TABLA 7. PUNTUACIONES SENSOR DE NIVEL DE PRESIÓN SONORA.....	34
TABLA 8. SENSORES SELECCIONADOS.....	34
TABLA 9. CARACTERÍSTICAS AMPLIFICADORES OPERACIONALES.....	35
TABLA 10. COMPONENTES UTILIZADOS EN EL CIRCUITO <i>HARVESTER</i> .....	44
TABLA 11. MEDIDAS OBTENIDAS EN LAS PRUEBAS DE IMPLEMENTACIÓN DEL <i>ENERGY HARVESTING</i> .....	46
TABLA 12. CÓDIGOS IDENTIFICATIVOS DE LAS VARIABLES .....	95
TABLA 13. PINES ASIGNADOS DEL MICROCONTROLADOR .....	97





## Capítulo 1. Introducción y objetivos.

### 1.1. Introducción.

Este proyecto surge de la necesidad de captar datos de variables ambientales con el objetivo de estudiar cómo influyen en la generación de ataques migrañosos e intentar prevenir posibles ataques.

#### 1.1.1. Motivación: La influencia de los parámetros ambientales en los enfermos de migraña.

##### 1.1.1.1. La migraña.

La migraña es un trastorno neurológico que afecta a entre un 10% y un 15% de la población europea según el estudio Epidemiology and Economic Impact of Migraine [1], siendo una enfermedad predominante en mujeres, las cuales representan el 80% de los afectados. En cuanto a la edad, aunque puede iniciarse a cualquier edad, generalmente los síntomas empiezan a mostrarse en personas entre 10 y 30 años<sup>1</sup>.

Hay diversos tipos de cefaleas con síntomas bien diferenciados que no hay que confundir con las migrañas. En la siguiente figura (Ilustración 1) se distingue la migraña de otros tipos de cefalea, causados por otras razones.



Ilustración 1. Tipos de cefalea

La migraña se presenta como dolor en un lado de la cabeza (hemicraneal) o en ambos lados de cabeza (holocraneal), y acompañado la mayoría de las veces de náuseas, vómitos, y mayor sensibilidad a la luz. Alrededor del 20%<sup>2</sup> de los pacientes sufren otros síntomas como depresión, irritabilidad, inquietud, náuseas o falta de apetito. En las migrañas, el sistema nervioso autónomo (SNA) regula la frecuencia cardíaca y respiratoria, sudoración y la actividad vasomotora, entre otras señales biomédicas. Sin embargo, todavía hay muchas incógnitas acerca de cómo la disautonomía de un paciente de migraña se ve afectada, y si es la causa o una consecuencia. En pacientes con aura (el paciente experimenta trastornos de percepción antes que el dolor) la llegada de dolor futuro puede ser asegurada; en pacientes con síntomas prodrómicos (perturbación subjetiva y no específica de la percepción) la aparición de migraña

<sup>1</sup> <http://www.aao.org/> Última vez consultada: 15/05/2016

<sup>2</sup> <http://www.aao.org/> Última vez consultada: 15/05/2016

puede no ocurrir. La anterioridad con la que aparecen estos síntomas prodrómicos es variable y desconocido, ya que los síntomas pueden aparecer en cualquier momento de 48 a 6 horas antes de la aparición de la migraña. Esta predicción es casi inútil para los propósitos de tratamiento (debido a la farmacocinética).

Las fases de la migraña se pueden dividir en cuatro fases<sup>3</sup>.

1. **Los síntomas premonitorios.** Estos síntomas son, por ejemplo, el cansancio, la dificultad para concentrarse, el aumento del apetito, bostezos o la retención excesiva de líquidos. Pueden aparecer incluso antes de dos horas del inicio del dolor, y se dan en más de la mitad, de los pacientes estudiados.

Psicológicos	Neurológicos	Generales
<b>Inhibitorios:</b> Depresión Lentitud mental Astenia  <b>Excitatorios:</b> Inquietud Actitud obsesiva Hiperactividad psicomotriz Euforia Ingeniosidad Irritabilidad Somnolencia	<b>Inhibitorios:</b> Dificultad de concentración  <b>Excitatorios:</b> Fotofobia Fonofobia Hiperosmia Bostezos	<b>Inhibitorios:</b> Sensación de frío Anorexia Astenia Estreñimiento Flatulencia  <b>Excitatorios:</b> Rigidez de cuello Ansiedad por alimentos Diarrea Sed Hiperactividad urinaria Retención hídrica

Ilustración 2. Síntomas prodrómicos<sup>4</sup>

2. **El aura.** Son síntomas que se van instaurando de forma progresiva. En la mayoría de los casos provocan alteraciones en la visión. El aura sólo se manifiesta en dos de cada diez personas y suele tener una duración, como ya se ha comentado, de entre diez y treinta minutos antes de que ocurra el dolor. Existen distintos tipos de aura, los más comunes, son los siguientes:

Visual	— Escotomas. — Fotopsias o destellos luminosos persistentes. — Fortificación espectral. — Movimiento de objetos: rotación oscilación o brillo. — Deslumbramiento. — Escotomas negativos. — Reducción del campo visual.
Alucinaciones visuales o distorsiones	— Distorsión de los objetos. — Macropsia. — Visión en mosaico. — Visión en zoom.
Sensitivas	— Parestesias. — Alteraciones olfatorias. — Alucinaciones.
Motoras	— Debilidad. — Ataxia.
Lenguaje	— Disartria. — Afasia global y sensitiva o motora.

Ilustración 3. Manifestaciones clínicas<sup>5</sup>

3. **Fase del dolor.** El tipo de dolor y su zona afectada varía en cada paciente pero suele manifestarse como una molestia, no muy intensa, que se va incrementando. La duración de la fase de dolor varía entre 4 y 72 horas. Aunque la mayor parte de los casos (cinco de cada seis) no llega a las 48 horas.

<sup>3</sup> <http://www.dmedicina.com/> Última vez consultada: 15/05/2016

<sup>4</sup> [https://www.cgcom.es/sites/default/files/guia\\_cefaleas.pdf](https://www.cgcom.es/sites/default/files/guia_cefaleas.pdf) Última vez consultada: 15/05/2016

<sup>5</sup> [https://www.cgcom.es/sites/default/files/guia\\_cefaleas.pdf](https://www.cgcom.es/sites/default/files/guia_cefaleas.pdf) Última vez consultada: 15/05/2016



4. **Fase de resolución o “resaca”.** En esta fase se encuentra la generalizada sensación de malestar que se queda después del intenso dolor.

Las causas concretas desencadenantes de las migrañas se desconocen, aunque actualmente se cree que la migraña es un trastorno constitucional con base genética, cuyos desencadenantes más comunes son:

- Hormonales, como por ejemplo la menstruación en mujeres.
- Alimenticios, como es la ingesta de alcohol o el consumo de cafeína.
- Estrés, tanto físico como psicológico, o la alteración del sueño.
- Debidos al entorno, como el exceso de luz, cambios climáticos, o baños calientes.

De entre los factores asociados a la aparición de migrañas destacar que:

- **Edad:** Como ya se ha mencionado anteriormente, la migraña es más propensa en personas de entre 10 y 30 años. Pero en la infancia, las migrañas afectan en el mismo nivel a niñas y niños, ya que es posteriormente cuando, a causa de los cambios hormonales, afecta más a las mujeres.
- **Herencia.** No está establecida completamente la relación aunque habitualmente las personas que son diagnosticadas de migrañas tienen antecedentes familiares.

Esto significa que, variables como la humedad relativa en el ambiente o la luz a la que se ve sometido el paciente pueden facilitar ataques de migraña. A partir de dichas variables se llevará a cabo el desarrollo de este proyecto, con la creación de una aplicación capaz de medir la humedad, luz, presión y temperatura para poder así, estudiar la relación entre las variaciones de estas variables y los ataques de migraña para poder detectar, e incluso predecir, los ataques de migrañas causadas por estos factores.

#### *1.1.1.2. El proyecto de investigación.*

Esta iniciativa se encuentra enmarcada dentro de un proyecto del grupo de investigación GreenDisc<sup>6,7</sup> de la Universidad Complutense de Madrid y de la Unidad de Cefaleas del Servicio de Neurología del Hospital Universitario de la Princesa de Madrid, con los que colabora el Grupo de Displays y Aplicaciones Fotónicas de la Universidad Carlos III de Madrid (GDAF-UC3M).

En el marco de este proyecto se ha diseñado un dispositivo inalámbrico para monitorización ambulatoria y de bajo coste que, mediante la monitorización de una serie de variables hemodinámicas, es capaz de predecir la aparición de migrañas aunque el paciente no experimente síntomas. La herramienta ha generado modelos de predicción que se anticipan a la aparición del dolor 47 minutos.

Las variables recogidas se sincronizan con las fases de dolor sufridas por los pacientes durante el ataque de migraña. A través de una metodología y el uso de técnicas algorítmicas de modelado predictivo, se relacionan estas variables con el dolor del paciente. Las técnicas de modelado predictivo que hasta ahora se han aplicado son: algoritmos de espacios de estados (algoritmo N4SID) [2] y técnicas de computación genética (gramáticas evolutivas) [3].

<sup>6</sup> <http://greendisc.dacya.ucm.es/> Última vez consultada 16/06/16

<sup>7</sup> <http://www.agenciasinc.es/Noticias/Un-modelo-piloto-predice-la-aparicion-de-migranas-incluso-sin-sintomas> Última vez consultada: 15/05/2016

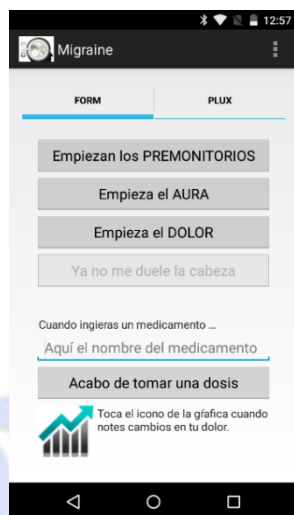


Ilustración 4. Aplicación Migraine

Como se puede ver en la imagen anterior (Ilustración 4) la aplicación dispone de varios botones que facilitan al usuario el manejo de la misma, permitiendo indicar en cada momento que tipo de síntoma está sintiendo o si toma algún medicamento.

El primer botón de la aplicación, “Empiezan los PREMONITORIOS”, registra el inicio de la fase de pródromo, explicada previamente. Los siguientes dos botones permiten al paciente indicar en qué fase se encuentra del ataque, ya sea el comienzo el aura, del dolor de cabeza en sí o si ha dejado de sentir la migraña. Cómo durante el proceso el dolor va cambiando, se ha añadido un botón para que el paciente indique la evolución de su dolor (Ilustración 5). Con esa curva sintomática subjetiva, se crea un modelo objetivo de curva de dolor que se ajusta a dos curvas semi-Gaussianas con las que posteriormente se entrenan los modelos de predicción.

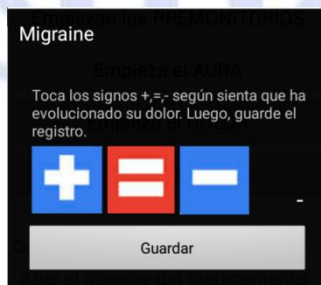


Ilustración 5. Indicación grados de dolor

Como continuación a este proyecto y con la idea de integrarla en la aplicación existente, este Trabajo Fin de Grado trata de diseñar un complemento que se pueda conectar con el teléfono por el conector de audio y que mida variables como la luz ambiente, la temperatura, humedad y ruido ambiente o nivel de presión sonora. Es importante que dicho circuito esté expuesto al ambiente para que pueda registrar todas las variables.

### 1.1.2. ¿Existe algún dispositivo comercial?

Existe un accesorio para el *Smartphone* de la empresa Shaka<sup>8</sup> (Ilustración 6) que permite monitorizar variables ambientales y que se conecta con el teléfono por el conector de audio. Está pensado principalmente para deportes náuticos, como la vela.

<sup>8</sup> <http://shakaon.net/products/> Última vez consultada: 15/05/2016





Ilustración 6. Accesorio marca Shaka

En esta versión, el accesorio es capaz de medir la dirección del viento, la velocidad, la temperatura ambiente y la presión con una precisión de  $\pm 3\%$  y  $\pm 1^\circ\text{C}$ ,  $\pm 3.0\%$  R.H y 0.4 kPa en lo referente al viento. Tiene un tamaño compacto y muy reducido, no llegando a los 10 gramos de peso, y su máxima dimensión son 6 cm de largo. Su precio en el mercado está alrededor de los 99 €.

Otras empresas también han sacado otros productos como el WeatherFlow<sup>9</sup>, en el caso de este accesorio únicamente mide la velocidad que tiene el viento y la dirección de este, con unas dimensiones muy similares a el anterior con un precio más asequible, 39.95 €.

Por último, también está Windoo (Ilustración 7 de la marca Skywatch<sup>10</sup>, sacando al mercado tres versiones del mismo accesorio, las cuales han numerado del 1 al 3, de la versión más simple a la versión más completa. Las funciones que tienen son velocidad del viento y temperatura, en su versión 1, añadiendo la medida de humedad relativa en la versión 2 y añadiendo las funciones como punto de rocío, presión atmosférica y altímetro.



Ilustración 7. Widoo

Su precio está en 85 € + I.V.A.

En la web de este producto, se informa de que este accesorio no tiene por qué funcionar correctamente en todos los dispositivos. Comunican que solo es compatible para unos modelos muy concretos que son los siguientes (Ilustración 8).

<sup>9</sup> [http://www.bhphotovideo.com/c/product/1187850-REG/weatherflow\\_wfano\\_01\\_windmeter\\_smatphone\\_weather\\_station.html](http://www.bhphotovideo.com/c/product/1187850-REG/weatherflow_wfano_01_windmeter_smatphone_weather_station.html) Última vez consultada: 15/05/2016

<sup>10</sup> <http://windoo.ch/> Última vez consultada: 15/05/2016



Ilustración 8. Dispositivos validos

Estas limitaciones pueden estar relacionadas a problemas de alimentación como se podrá ver en apartados posteriores (aparatado 2.3.1. Obtención de energía del teléfono., en particular).

Todas estas aplicaciones comerciales son *Plug and Play*. Se desconocen los datos de consumo de energía que requiere cada accesorio.

Por último, se tiene que comentar que cada una los accesorios que se han nombrado disponen de aplicaciones para el Smartphone propias. Las aplicaciones propietarias y los protocolos de comunicación entre el accesorio y la aplicación impiden al acceso a los datos brutos así como el control de la tasa de muestreo. Esto fundamenta la necesidad de crear un dispositivo propio para el estudio de predicción de migrañas.

## 1.2. Objetivos.

### 1.2.1. Objetivo Principal.

El objetivo principal de este proyecto es el diseño e implementación de una estación meteorológica portátil, para la integración en un estudio de afección de variables ambientales en la enfermedad de migraña.

### 1.2.2. Objetivos secundarios.

El proyecto de predicción de crisis de migraña limita sus modelos predictivos a cuatro variables hemodinámicas: temperatura superficial de la piel, sudoración, ritmo cardiaco y saturación de oxígeno. Los tiempos de predicción que se manejan y se intuyen que se pueden alcanzar, no superan 1 hora. Con el desarrollo de este TFG se pretende ampliar el estudio con el uso de variables ambientales para ayudar a la predicción basada en variables fisiológicas. Los datos ambientales que se quieren medir se obtendrán a lo largo del día de manera no intrusiva mientras dure la monitorización ambulatoria de variables hemodinámicas en el tiempo que dura el estudio. La idea es que el paciente pueda llevar el dispositivo siempre encima, permitiendo así un control constante de su estado, por lo que dicho dispositivo debe ser portable. Hoy en día, la tecnología acompaña a las personas en todo momento, ya sea en teléfonos móviles, tabletas, etc. Por ello que se considera que la opción para implementar el sistema desarrollado ha de pasar por el uso de un dispositivo móvil (en principio en entorno Android).

Por otro lado se tiene en cuenta el tamaño del dispositivo diseñado, ya que aunque no se pretende ser ergonómico, a la hora de transportar el dispositivo, reduciendo su tamaño, el paciente se mostrará menos reacio a emplearlo. Este es también un factor importante, ya que el estudio se realiza con pacientes voluntarios, y cuanto más cómodos estén el número de voluntarios será mayor y, por consecuencia, los datos obtenidos en el estudio se podrán contrastar mejor y serán más representativos.

Otro problema que se aborda en este TFG es el tema de la alimentación del dispositivo. Al ser un sistema portátil se debe encontrar la mejor forma de alimentar el dispositivo para aumentar al máximo su duración.

### 1.3. Especificaciones del sistema.

- **Toma de datos.** Se tomarán datos de cinco variables. Estas serán: luz ambiente, temperatura, humedad relativa, presión y nivel de presión sonora.
- **Comunicación.** Se quiere conseguir una comunicación entre el teléfono y la estación meteorológica. Se realizará a través del conector de audio.
- **Portátil.** Como ya se ha mencionado anteriormente, una buena opción para el objetivo de la portabilidad es que la estación esté conectada con el teléfono. Esta estación meteorológica, al estar conectada al teléfono y ser transportada por el paciente durante todo el día, se hace necesario encontrar una solución al problema de alimentación del dispositivo.
- **Alimentación y consumo.** De acuerdo con la línea de trabajo que tiene el grupo de investigación, se pretende que el consumo que tenga nuestro dispositivo sea bajo. Una de las opciones que se barajan es que el dispositivo sea alimentado por el propio teléfono. Actualmente, disponer de un Smartphone cuya batería dure lo máximo posible, alrededor de unas 16 o 18 horas, es el principal problema en la industria de estos dispositivos. Es por ello que en este proyecto se tratará en que el consumo afecte lo más mínimo al dispositivo móvil.
- **Almacenamiento.** Los datos son almacenados temporalmente y transmitidos a un servidor del grupo de investigación para su tratamiento y uso.
- **Tamaño.** Respecto al tamaño del sistema Smartphone-Estación meteorológica se pretende que sea lo más pequeño posible para seguir ofreciendo la comodidad de su manejo y su transporte al paciente. En cuanto a las medidas, se buscará que el sistema resultante no tenga unas dimensiones superiores a las del ancho del teléfono tanto en largo como en ancho del sistema final ya que si supera el ancho del teléfono podría resultar incómodo, al igual que si el dispositivo fuese muy largo. De esta manera podrá ir ligado al teléfono como si fuera uno. Se trabajará para no superar las dimensiones del ancho del teléfono.
- **Bajo coste.** En este proyecto, al igual que la mayoría de los sectores, el dinero es un factor siempre a tener en cuenta. Se intentará conseguir un diseño que sea lo menos costoso posible de cara a su futura producción.

## 1.4. Fases del proyecto.

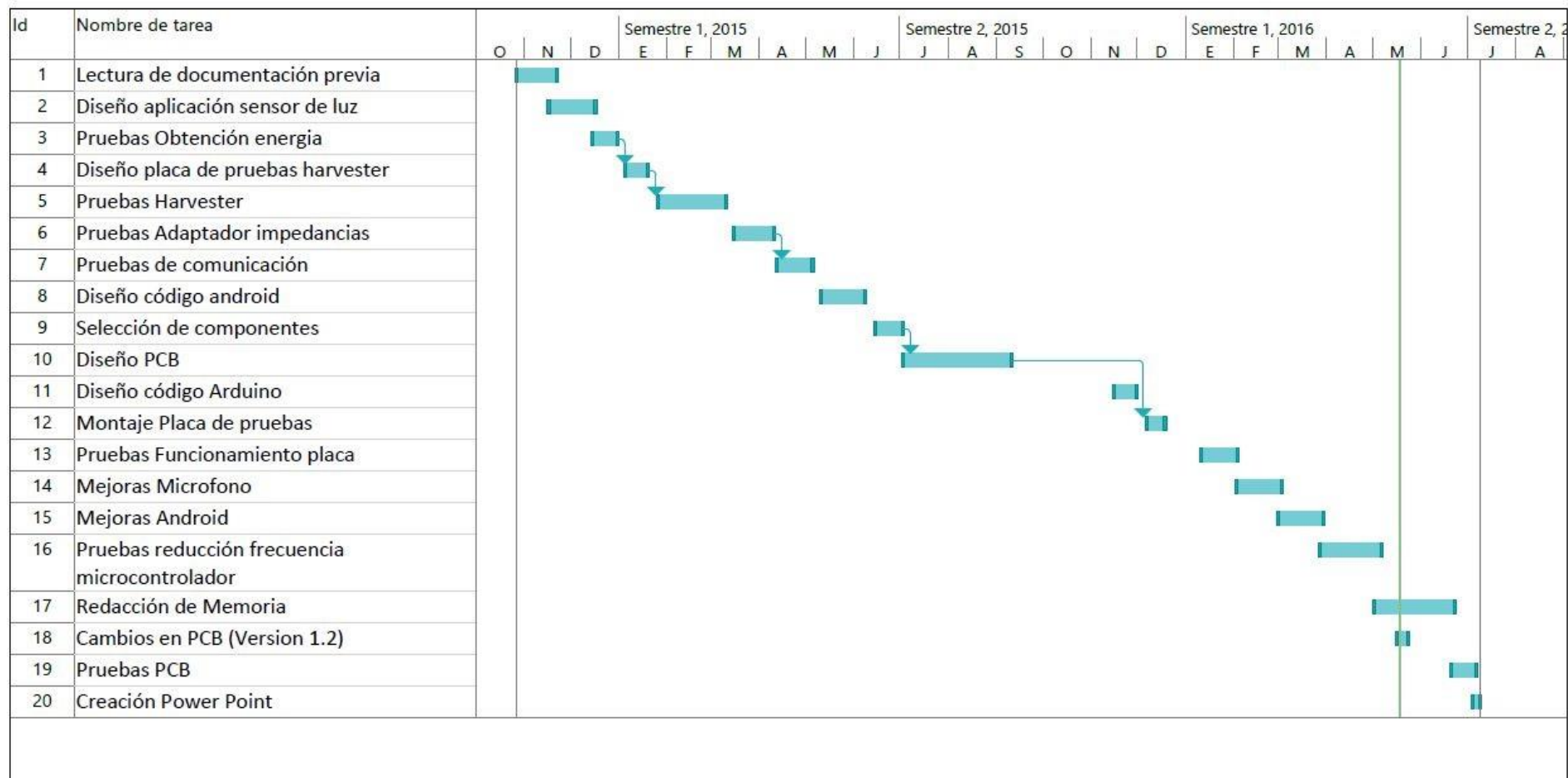


Ilustración 9. Diagrama de Gantt



## 1.5. Medios utilizados.

En el proyecto se ha trabajado con una gran cantidad de herramientas y medios. Por este motivo se van a separar en varios apartados.

### 1.5.1. Instalaciones.

El laboratorio del GDAF permite el acceso a las herramientas comunes de cualquier trabajo relacionado con la electrónica, tales como osciloscopio, generadores de funciones, componentes electrónicos, puestos de soldadura, etc.

### 1.5.2. Hardware.

También se han empleado herramientas de hardware para la realización del proyecto, en concreto vamos a hacer referencia a dos equipos destacados.

- Arduino Uno

El diseño de la estación meteorológica, estará centrado en el microcontrolador ATmega328P, que se encuentra integrado en el Arduino Uno. Por lo que usaremos esta placa para prototipado y pruebas.



Ilustración 10. Placa Arduino Uno

Arduino<sup>11</sup> es un entorno de hardware libre y de bajo coste, compuesto por una placa, con un microcontrolador y puertos de entrada y salida, lo que permite el desarrollo de forma sencilla de múltiples diseños. La placa es programada a través del ordenador al cual se conecta a través de un puerto USB, y de otro microprocesador, el ATmega16U2 que hace las veces de programador del ATmega328P.

Por otro lado, Arduino también se compone de un entorno de desarrollo, donde poder programar. Es más conocido como Integrated Development Environment (el IDE de Arduino), el cual se explicará más adelante.

De manera externa a la marca Arduino, el sector de venta de componentes electrónicos ha sabido captar el tirón generado por esta, y en los últimos años han comenzado a ofrecer módulos compatibles con Arduino que integran sensores de todo tipo a unos precios muy competitivos; esto ha provocado que los sensores en los que están basados esos módulos vean reducidos sus precios.

<sup>11</sup> <https://www.arduino.cc/> Última vez consultada: 15/05/2016

- Motorola Moto G

La estación meteorológica forma parte de un sistema complejo que concentra los datos en un teléfono móvil inteligente. En este estudio se utiliza el modelo Motorola Moto G<sup>12</sup>.

Algunas de las principales características del mismo se resumen a continuación:

- Acelerómetro, sensor de luz ambiente, magnetómetro, brújula, sensor de proximidad.
- 2 altavoces frontales y micrófono con cancelación de ruido.
- Batería de 2470 mAh, uso mixto hasta 24 horas.
- Lanzado con Android Lollipop versión 5.1.1, actualizable a la versión 6.0 Marshmallow.

Estas características permiten que se puedan usar algunos de los propios sensores ambientales que incluye el móvil para la aplicación que se desarrolla en este proyecto. En particular haremos uso del sensor de luz ambiente.

### 1.5.3. Herramientas de Software.

El software es una parte fundamental dentro del proyecto, una gran parte del desarrollo del mismo se ha hecho a nivel de software. Bien puede ser, desarrollando la aplicación móvil, diseñando PCB, sacando los *gerbers* de la PCB. Se han utilizado una gran variedad de programas que se van a explicar brevemente a continuación.

1. **Desarrollo de código para el firmware del microcontrolador ATmega328:**

Como ya se ha comentado anteriormente, la plataforma Arduino, dispone de un entorno de desarrollo desde la cual se puede programar la placa. En este caso concreto, se utilizará la versión 1.0.6., aunque la última a fecha de redacción de este documento es la 1.6.0. La versión 1.0.6. es la mínima que permite el manejo de las librerías de SoftModem que se explicarán más adelante.



Ilustración 11.  
Logo Arduino

2. **Android Studio para el desarrollo de la aplicación móvil** Android Studio es un entorno de desarrollo integrado para la plataforma Android. Está desarrollado por Google, y la primera versión estable surgió a finales del año 2014. Actualmente, se ha convertido en la primera herramienta de desarrollo para aplicaciones Android posicionándose por delante de Eclipse.



Ilustración 12. Logo  
Android Studio

3. **E.A.G.L.E. para el diseño de PCB (Printed Circuit Board).** Es un programa de diseño de PCBs. Es muy conocido por proyectos DIY (Do It Yourself) ya que una gran variedad de versiones de este programa tienen licencia Freeware. Es un programa con un uso muy intuitivo y con múltiples librerías de componentes electrónicos. Con este programa también se han generado los *gerbers* que después se utilizaran para la fabricación de la placa.



Ilustración 13.  
Logo EAGLE



Ilustración 14.  
Logo Gerberlogix

<sup>12</sup> <https://www.motorola.es/products/moto-g> Última vez consultada: 15/05/2016



4. **Gerberlogix para la visualización de *gerbers*.** Un *gerber* es un tipo de formato de archivo que contiene la información necesaria para la fabricación de circuitos impresos.
5. **Audacity para la generación de archivos de audio.** Dentro del proyecto, una necesidad importante ha sido el caracterizar convenientemente la funcionalidad del conector *jack* del móvil. Para ello se ha necesitado el generar tonos puros. Audacity es un software libre, que se puede usar para la grabación y edición de audio.



Ilustración 15. Logo Audacity

6. **Signal Generator para la reproducción de ondas por el altavoz.** Alternativamente, también se ha utilizado esta aplicación Android de ZYZ-Apps. La misma permite generar ondas de tipo senoidal, cuadrada, triangular y diente de sierra, y variar la frecuencia y su amplitud.



Ilustración 16. Logo Signal Generator

7. **SketchUp para el diseño 3D.** Durante este proyecto se han realizado diseños para la creación de una carcasa que recubra la estación meteorológica y la proteja de golpes.



Ilustración 17. Logo SketchUp



## Capítulo 2. Diseño del sistema hardware.

### 2.1. Diseño del sistema completo.

Las partes de las que se compone este proyecto se pueden diferenciar según el siguiente diagrama.

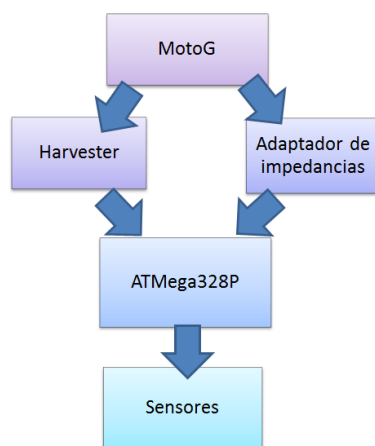


Ilustración 18. Diagrama de funcionamiento del sistema

El sistema se conecta al teléfono mediante un conector *jack*. La conexión con el móvil se utiliza para dos funcionalidades: proporcionar la fuente de energía al circuito *harvester* (traducido del inglés: recolector), el cual es un circuito que se encarga de “recolectar” la energía que le proporciona el teléfono (a través de una onda senoidal que éste convierte en una tensión continua), y otra funcionalidad destinada a la comunicación entre Arduino y el teléfono, tratando de adaptar las impedancias de entrada y salida.

El microcontrolador se encarga de comunicarse con los sensores y recolectar la información de las variables ambientales. Dichas mediciones se realizan de manera secuencial por cada sensor: uno de humedad, otro de presión y temperatura, según corresponda.

El sistema completo del hardware que se ha realizado en este proyecto se muestra en el [Anexo 1.1](#), en el cual se diferencian varios subsistemas.

En dicho esquema podemos diferenciar:

- un convertidor continua-continua, de una alta eficiencia, pensado para la regular la tensión de salida y fijarla a un valor deseado entre el circuito *harvester* y el microcontrolador.
- los circuitos de alimentación de los sensores. Estos sensores se alimentan a través de un transistor de corte para tener solo uno activo a la vez.
- la adaptación de impedancias entre el teléfono móvil y el microcontrolador en ambas direcciones. Existen dos canales, el canal en el cual el teléfono se comunica con el microcontrolador (FSKIN) y el canal en el que el microcontrolador se comunica con el teléfono (FSKOUT).
- Por último el circuito *harvester*, al que se le ha añadido un amplificador operacional como prueba para realizar la adaptación de impedancias.

La parte del circuito que no tiene un área marcada representa el ATMega328, y los componentes necesarios para su funcionamiento. También se le ha añadido en el puerto 19

del ATmega328, que correspondería con el pin digital 13 de la placa Arduino, un led para poder testear el funcionamiento del microcontrolador.

## 2.2. Diseño del sistema hardware.

A continuación, se diseña todo el circuito de adquisición de datos y se seleccionan todos los componentes principales y secundarios necesarios para su funcionamiento.

### 2.2.1. Requisitos y características detalladas de los sensores.

Existen una serie de requisitos generales que se pueden aplicar a todos los componentes que se van a introducir en la estación meteorológica:

- *Tamaño:* Todos los componentes seleccionados para este dispositivo deben tener el menor tamaño posible. El tamaño de la placa final puede verse muy afectado por esta variable. Esto exige que los sensores seleccionados tengan un encapsulado SMD.
- *Consumo:* En este dispositivo, la alimentación, como ya se ha visto en apartados anteriores es un tema importante a tratar. Aquellos componentes que sean seleccionados deben consumir lo menos posible.
- *Tipo de soldadura:* Los componentes SMD deben ser soldados en hornos especiales a los que no tenemos acceso. Por esta razón los componentes han sido escogido tales que los pines de conexión sean accesibles para una soldadura manual.
- *Precio:* Es un requisito habitual. No restrictivo pero se debe buscar componentes económicos para nuestro dispositivo.
- *Puertos:* de cara a la simplicidad del diseño se dará prioridad a sensores con interfaz I2C.

El microcontrolador que se va a utilizar para gestionar los sensores y comunicarse con ellos será el ATMEGA328P-PU. Sus características son las siguientes:

- 20 MHz Frecuencia Operativa Máxima
- 32 KB Memoria Flash
- 2 KB Memoria SRAM
- 32 Registros de propósito general
- 23 GPIO
- Oscilador interno

El microcontrolador seleccionado tiene un encapsulado DIP, ya que se instala en una placa de pruebas. En la versión definitiva del dispositivo se cambiará a un encapsulado SMD aunque en este proyecto no ha sido posible.

La presión atmosférica en condiciones normales suele oscilar entre 900 milibares y 1100 milibares, por lo que el sensor seleccionado debe poder cubrir ese rango de presiones. El sensor debe poder funcionar en las condiciones normales mencionadas. Y debe tener una mínima resolución de 1 milibar para poder ver los cambios de presión a lo largo del día.

La temperatura ambiente en un día normal salvando las temperaturas extremas, se puede establecer un límite inferior de 0 grados centígrados. Es cierto que durante el invierno se alcanzan temperaturas inferiores, pudiendo llegar a -5 grados centígrados, pero esas temperaturas ocurren a horas de madrugada poco habituales para que los pacientes las sufran.



Por otro lado, las temperaturas superiores se pueden establecer en 45 grados centígrados, pudiendo ser alcanzadas en los meses de verano. En cuanto la precisión de la medida, sería útil captar variaciones de 0.1 grados.

En la atmósfera, los ambientes en los que se moverá el paciente serán ambientes poco extremos. Ambientes relativamente secos, con una humedad en torno al 15% o 20%. Por el contrario, en momentos de tormenta, cuando llueve, la humedad relativa aumente mucho, hasta niveles de 85-90%. La precisión de estas mediciones sería correcta con una unidad de %R.H.

En cuanto a la variable lumínica, los valores mínimos que se pueden dar, sin fuentes de luz durante la noche, pueden ser 0 lux. Por el contrario en situaciones con buena luz, como en oficinas, se recomienda tener una luz de 500 lux, siendo óptimo los 750 lux. Una precisión correcta sería poder medir las variaciones de una unidad de lux.

En cuanto al audio, el sensor que se implemente debe ser capaz de medir ruido desde un susurro que supondría unos 15 dB, a un ruido muy alto que pudiera producirse, alrededor de unos 90 dB, con una resolución de 1dB. El sensor de presión no nos proporcionara el dato en decibelios directamente, por lo que habrá que tratar la señal que proporcione el sensor.

### 2.2.2. Selección.

Una vez conocidos los requisitos que se van a necesitar de los sensores, se lleva a cabo una búsqueda de distintos posibles dispositivos. A continuación los criterios que se van a seguir para la selección de cada sesión.

Ya que cada grupo de componentes puede tener unas características diferentes, por ejemplo los sensores de presión sonora tienen unos rangos de consumo menores que el resto de sensores, por lo que para cada apartado evaluado de cada sensor se evaluarán se establecerán unos márgenes para establecer la puntuación. Por cada limite que supere de manera positiva se otorgará un punto, salvo en el caso del consumo de los componentes, que se otorgará 3 puntos. Esto es debido a que en todo momento en este proyecto se ha buscado componentes *low power*. Además también se otorgarán puntos adicionales a los sensores que posean características especiales, como por ejemplo si un sensor es capaz de medir dos variables, lo que permitiría reducir componentes. Por último hay que comentar que se otorgara un punto a aquellos sensores cuyo tipo de encapsulado no requiera una soldadura por horno. La siguiente tabla resume los criterios de evaluación que se seguirán.

Puntos (x3 en caso de consumo)		0	1	2	3	4	5
<b>Presión</b>	Consumo (mA)	>5	5-2	2-1	1-0.7	0.7-0.5	<0.5
	Precio (€)	>5	5-3	3-2.5	2.5-2	<2	
	Tamaño (mm <sup>2</sup> )	>20	20-15	15-12	12-10	<10	
	Soldadura	Horno	No horno				
	Variable Extra	No		Si			
<b>Humedad</b>	Consumo (mA)	>1	1-0.7	0.7-0.5	0.5-0.3	<0.3	
	Precio (€)	>15	15-10	10-5	5-3	<3	
	Tamaño (mm <sup>2</sup> )	>50	50-20	20-10	<10		
	Soldadura	Horno	No horno				
	Extra	No		Si			
<b>Presión sonora</b>	Consumo (μA)	>500	500-200	200-100	<100		
	Precio (€)	>2.5	2.5-2	2-1.5	<1.5		
	Sensibilidad(dB)	<-50	-50 – -40	-40 – -30	-30 – -20	>-20	
	Tamaño	>20	20-15	15-10	<10		

Tabla 1. Puntuaciones de los sensores

## 1. Presión

A continuación se muestran los sensores de presión considerados, así como sus características y la puntuación asignada según lo anteriormente descrito.

 <b>MPL3115A2</b>  Ilustración 19. MPL3115A2	<ul style="list-style-type: none"> <li>• Tipo de Presión Absoluta</li> <li>• Presión Operativa Mín. 20 kPa</li> <li>• Presión Operativa Máx. 110 kPa</li> <li>• Tensión de Alimentación Mín. 1.95 V</li> <li>• Tensión de Alimentación Máx. 3.6 V</li> <li>• Número de Contactos Macho 8</li> </ul>	<ul style="list-style-type: none"> <li>• Número de Contactos Macho 8</li> <li>• Temperatura de Trabajo Mín. -40 °C</li> <li>• Temperatura de Trabajo Máx. 85 °C</li> <li>• Corriente durante la medición: 2 mA</li> <li>• Precio: 2.98€</li> <li>• Dimensiones: 5.0 mm x 3.0 mm x 1.2 mm</li> </ul>
 <b>BMP280</b>  Ilustración 20. BMP 280	<ul style="list-style-type: none"> <li>• Tipo de Presión. Absoluta.</li> <li>• Presión Operativa Mín. 30hPa.</li> <li>• Presión Operativa Máx. 110hPa.</li> <li>• Resolución Presión: 0.01hPa.</li> <li>• Temperatura Operativa Mín.: 0 °C.</li> <li>• Temperatura Operativa Max: 65 °C.</li> <li>• Resolución Temperatura: 0.1 °C.</li> <li>• Tensión de Alimentación Mín. 1.71 V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6 V.</li> <li>• Número de Contactos Macho. 8.</li> <li>• Temperatura de Trabajo Mín. -40 °C.</li> <li>• Temperatura de Trabajo Máx. 85 °C.</li> <li>• Corriente durante la medición. 0.72 mA.</li> <li>• Precio: 2.74 €.</li> <li>• Dimensiones: 2.0 mm x 2.5 mm x 0.9 mm.</li> </ul>
 <b>MS563702BA03-50</b>  Ilustración 21. MS563702BA03-50	<ul style="list-style-type: none"> <li>• Tipo de Presión. Barométrica.</li> <li>• Presión Operativa Mín. 300 mbar.</li> <li>• Presión Operativa Máx. 1200 mbar.</li> <li>• Tensión de Alimentación Mín. 1.8 V.</li> <li>• Tensión de Alimentación Máx. 3.6 V.</li> <li>• Sensor, Tipo. QFN.</li> <li>• Número de Contactos Macho. 8.</li> </ul>	<ul style="list-style-type: none"> <li>• Temperatura de Trabajo Mín. -40 °C.</li> <li>• Temperatura de Trabajo Máx. 85 °C.</li> <li>• Corriente durante la medición: 1.25 mA.</li> <li>• Precio: 2.88 €.</li> <li>• Dimensiones: 3.0 mm x 3.0 mm x 0.9 mm.</li> </ul>
 <b>MPX5100</b>  Ilustración 22. MPX5100	<ul style="list-style-type: none"> <li>• Tipo de Presión. Absoluta.</li> <li>• Sensibilidad, V/P. 45 mV/kPa.</li> <li>• Presión Operativa Mín. 15 kPa.</li> <li>• Presión Operativa Máx. 115 kPa.</li> <li>• Tensión de Alimentación Mín. 4.75 V.</li> <li>• Tensión de Alimentación Máx. 5.25 V.</li> <li>• Sensor, Tipo. SIP.</li> </ul>	<ul style="list-style-type: none"> <li>• Número de Contactos Macho. 6.</li> <li>• Temperatura de Trabajo Mín. -40 °C.</li> <li>• Temperatura de Trabajo Máx. 125 °C.</li> <li>• Corriente: 7 mA.</li> <li>• Dimensiones: 11.3 mm x 18.0 mm x 13.0 mm.</li> <li>• Precio: 14,55 €.</li> </ul>
 <b>BMP180</b>  Ilustración 23. BMP 180	<ul style="list-style-type: none"> <li>• Tipo de Presión. Absoluta.</li> <li>• Presión Operativa Mín. 30 hPa.</li> <li>• Presión Operativa Máx. 110 hPa.</li> <li>• Resolución Presión: 0.01 hPa.</li> <li>• Temperatura Operativa Mín.: 0 °C.</li> <li>• Temperatura Operativa Max: 65 °C.</li> <li>• Resolución Temperatura: 0.1 °C.</li> <li>• Tensión de Alimentación Mín. 1.71 V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6 V.</li> <li>• Número de Contactos Macho. 8.</li> <li>• Temperatura de Trabajo Mín. -40 °C.</li> <li>• Temperatura de Trabajo Máx. 85 °C.</li> <li>• Corriente durante la medición 0.65 mA.</li> <li>• Precio: 3.73 €.</li> <li>• Dimensiones: 3.35 mm x 3.35 mm x 0.93 mm.</li> </ul>

Tabla 2. Características sensores de presión

Sensor	Consumo	Precio	Tamaño	Soldadura	Variable extra	Total
MPL3115A2	3	2	2	0	0	7
BMP280	9	2	4	0	2	17
MS563702BA03-50	6	2	4	0	0	12
MPX5100	0	0	0	1	0	1
BMP180	12	1	3	0	2	18

Tabla 3. Puntuaciones sensor de presión



## 2. Humedad

Los sensores de humedad analizados y sus principales parámetros se resumen en las siguientes tablas.

 <p>Ilustración 24. CHS-MSS</p>	<ul style="list-style-type: none"> <li>• Precisión de Detección. 5%.</li> <li>• Rango de Humedad. 20% a 85%.</li> <li>• Salida del Sensor. Tensión.</li> <li>• Sensor, Terminales. Agujero Pasante.</li> <li>• Tiempo de Respuesta. 1min.</li> <li>• Tensión de Alimentación Mín. 4.75VDC.</li> </ul>	<ul style="list-style-type: none"> <li>• Temperatura de Trabajo Mín. 5°C.</li> <li>• Temperatura de Trabajo Máx. 45°C.</li> <li>• Salida 10mV/%RH.</li> <li>• Corriente de Alimentación 600 µA.</li> <li>• Precio: 17.24€.</li> <li>• Dimensiones: 20mm x 10 mm x 5 mm.</li> <li>• Tensión de Alimentación Máx. 5.25VDC.</li> </ul>
 <p>Ilustración 25. HTU21D</p>	<ul style="list-style-type: none"> <li>• Precisión de Detección: 5%.</li> <li>• Rango de Humedad: 0% a 100%.</li> <li>• Salida del Sensor: Digital.</li> <li>• Sensor, Terminales: SMD/SMT.</li> <li>• Resolución: 12bits.</li> <li>• Tensión de Alimentación Mín. 1.5VDC.</li> </ul>	<ul style="list-style-type: none"> <li>• Temperatura de Trabajo Mín. -40°C.</li> <li>• Temperatura de Trabajo Máx. 125°C.</li> <li>• Resolución de temperatura 0.01 °C.</li> <li>• Corriente de Alimentación 500 µA.</li> <li>• Precio: 9.98€.</li> <li>• Dimensiones: 3mm x 3 mm x 0.9 mm.</li> <li>• Tensión de Alimentación Máx. 3.6VDC.</li> </ul>
 <p>Ilustración 26. SHT21</p>	<ul style="list-style-type: none"> <li>• Precisión de Detección: 0.04%.</li> <li>• Rango de Humedad: 0% a 100%</li> <li>• Salida del Sensor: Digital.</li> <li>• Sensor, Terminales: SMD/SMT</li> <li>• Resolución: 12bits.</li> <li>• Tensión de Alimentación Mín. 2.1VDC</li> </ul>	<ul style="list-style-type: none"> <li>• Temperatura de Trabajo Mín. -40°C</li> <li>• Temperatura de Trabajo Máx. 125°C</li> <li>• Resolución de temperatura 0.01 °C.</li> <li>• Corriente de Alimentación: 300 µA.</li> <li>• Precio: 3.66€.</li> <li>• Dimensiones: 3mm x 3 mm x 1.1 mm.</li> <li>• Tensión de Alimentación Máx. 3.6VDC</li> </ul>
 <p>Ilustración 27. ChipCap 2</p>	<ul style="list-style-type: none"> <li>• Precisión de Detección: 0.01% R.H.</li> <li>• Rango de Humedad: 0% a 100%.</li> <li>• Salida del Sensor: Digital.</li> <li>• Sensor, Terminales: SMD/SMT.</li> <li>• Resolución: 14bits.</li> <li>• Tensión de Alimentación Mín. 2.3VDC.</li> </ul>	<ul style="list-style-type: none"> <li>• Temperatura de Trabajo Mín. -40°C.</li> <li>• Temperatura de Trabajo Máx. 125°C.</li> <li>• Resolución de temperatura 0.01 °C.</li> <li>• Corriente de Alimentación: 750 µA.</li> <li>• Precio: 12.35€.</li> <li>• Dimensiones: 6mm x 4 mm x 1.9 mm.</li> <li>• Tensión de Alimentación Máx. 5.5VDC.</li> </ul>

Tabla 4. Características sensores de humedad

Sensor	Consumo	Precio	Tamaño	Variable extra	Soldadura	Total
CHS-MSS	6	0	0	0	1	7
HTU21D	9	2	3	2	0	17
SHT21	12	3	3	2	0	21
ChipCap 2	3	1	2	2	1	10

Tabla 5. Puntuaciones sensor de humedad

## 3. Presión sonora

Como el en dispositivo móvil va a estar constantemente conectado al conector *jack*, el Sistema operativo Android deshabilita el micrófono ubicado en el teléfono, por este motivo se necesita instalar un micrófono en la placa. Todos los componentes visualizados para evaluar la presión sonora, necesitan una amplificación posterior a la medida tomada. Por esta razón, en

este apartado, se van a evaluar dos amplificadores de bajo consumo para añadir uno de ellos al diseño final de nuestra placa.

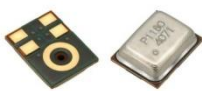
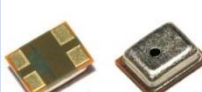
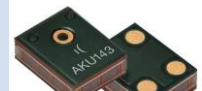
	<ul style="list-style-type: none"> <li>• Transductor, Función: Microphone.</li> <li>• Directividad: Omnidireccional.</li> <li>• Respuesta en Frecuencia Mín. 100 Hz.</li> <li>• Respuesta en Frecuencia Máx. 10 kHz.</li> <li>• Tensión de Alimentación Mín. 1.6V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6V.</li> <li>• Sensibilidad (dB): -26dB.</li> <li>• Intensidad: 155 <math>\mu</math>A.</li> <li>• Precio: 1.79€.</li> <li>• Dimensiones: 3.76mm x 2.95mm x 1.10 mm 11.092.</li> </ul>
	<ul style="list-style-type: none"> <li>• Transductor, Función: Microphone.</li> <li>• Directividad: Omnidireccional.</li> <li>• Respuesta en Frecuencia Mín. 100 Hz.</li> <li>• Respuesta en Frecuencia Máx. 10 kHz.</li> <li>• Tensión de Alimentación Mín. 1.5V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6V.</li> <li>• Sensibilidad (dB): -42dB.</li> <li>• Intensidad: 75<math>\mu</math>A.</li> <li>• Precio: 1.02€.</li> <li>• Dimensiones: 3.5mm x 2.65mm x 0.98 mm 9.275.</li> </ul>
	<ul style="list-style-type: none"> <li>• Transductor, Función: Microphone.</li> <li>• Directividad: Omnidireccional.</li> <li>• Respuesta en Frecuencia Mín. 100 Hz.</li> <li>• Respuesta en Frecuencia Máx. 10 kHz.</li> <li>• Tensión de Alimentación Mín. 1.6V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6V.</li> <li>• Sensibilidad (dB): -26dB.</li> <li>• Intensidad: 620<math>\mu</math>A.</li> <li>• Precio: 1.71€.</li> <li>• Dimensiones: 3.5mm x 2.65mm x 0.98 mm 9.275.</li> </ul>
	<ul style="list-style-type: none"> <li>• Transductor, Función: Microphone.</li> <li>• Directividad: Omnidireccional.</li> <li>• Respuesta en Frecuencia Mín. 50 Hz.</li> <li>• Respuesta en Frecuencia Máx. 11.5 kHz.</li> <li>• Tensión de Alimentación Mín. 1.6V.</li> </ul>	<ul style="list-style-type: none"> <li>• Tensión de Alimentación Máx. 3.6V.</li> <li>• Sensibilidad (dB): -42dB.</li> <li>• Intensidad: 155 <math>\mu</math>A.</li> <li>• Precio: 2.11€.</li> <li>• Dimensiones: 3.76mm x 2.95mm x 1.10 mm 11.092.</li> </ul>

Tabla 6. Características sensores de nivel de presión sonora

Sensor	Consumo	Tamaño	Sensibilidad	Precio	Total
SPH0641LM4H1	6	2	3	2	13
SPW2430HR5HB	9	3	1	3	16
SPH0642HT5H-1	0	3	3	2	8
AKU143	6	2	1	1	10

Tabla 7. Puntuaciones sensor de nivel de presión sonora.

Finalmente, de acuerdo a las puntuaciones los sensores seleccionados son los siguientes.

	Presión	Humedad	Presión sonora
Sensor	BMP180	SHT21	SPW2430HR5HB

Tabla 8. Sensores seleccionados.

Ahora se procederá a realizar una comparación entre las características de dos amplificadores operacionales *low power* que se han buscado para incluir en nuestro diseño. Se seleccionará el que mejores características posea.

TSV621AILT	MCP6001
<ul style="list-style-type: none"> <li>● Ancho de Banda: 420kHz.</li> <li>● Núm. de Amplificadores: 1.</li> <li>● Velocidad de Cambio (Slew Rate): 0.14V/μs.</li> <li>● Tensión de Alimentación, Rango: 1.5V a 5.5V.</li> <li>● Tipo de Amplificador: SOT-23.</li> <li>● Núm. de Contactos Macho: 5.</li> <li>● Corriente: 38 μA.</li> <li>● Precio: 0.66€.</li> <li>● Dimensiones: 2.9mm x 2.8mm x 1.2 mm.</li> </ul>	<ul style="list-style-type: none"> <li>● Ancho de Banda: 1MHz.</li> <li>● Núm. de Amplificadores: 1.</li> <li>● Velocidad de Cambio (Slew Rate): 0.6V/μs.</li> <li>● Tensión de Alimentación, Rango: 1.8V a 6V.</li> <li>● Tipo de Amplificador: SOT-23.</li> <li>● Núm. de Contactos Macho: 5.</li> <li>● Corriente: 100 μA.</li> <li>● Precio: 0.26€.</li> <li>● Dimensiones: 2.9mm x 2.8mm x 1.2 mm.</li> </ul>

Tabla 9. Características amplificadores operacionales.

Ya que ambos poseen el mismo tamaño de huella y los mismos rangos de tensión de alimentación, las variables evaluables que se tienen en cuenta son el precio y el consumo. Debido a la importancia de este último, se acaba eligiendo el TSV621AILT.

### 2.2.2.1. Temperatura.

Como se ha visto en los apartados anteriores, los sensores seleccionados son capaces de medir la temperatura con una resolución que cumple los requisitos expuestos. De esta forma se evita tener que añadir un sensor específico solo para poder realizar dicha medición ahorrando espacio en la placa, se usará el que dispone el sensor BMP180.

### 2.2.2.2. Luz.

Para esta variable no se va a seleccionar ningún componente. Como se verá en apartados posteriores, se ha conseguido capturar de otra forma siendo sustituido por una función del teléfono móvil. La medición de los luxes será realizada dentro de la propia aplicación final de Android, no se requerirá que se capte externamente.

### 2.2.3. Esquema general del sistema sensor.

Finalmente, una vez elegidos los sensores que formaran la estación meteorológica, el esquemático de la parte de hardware referente a los sensores es la que se muestra en la figura siguiente (Ilustración 32).

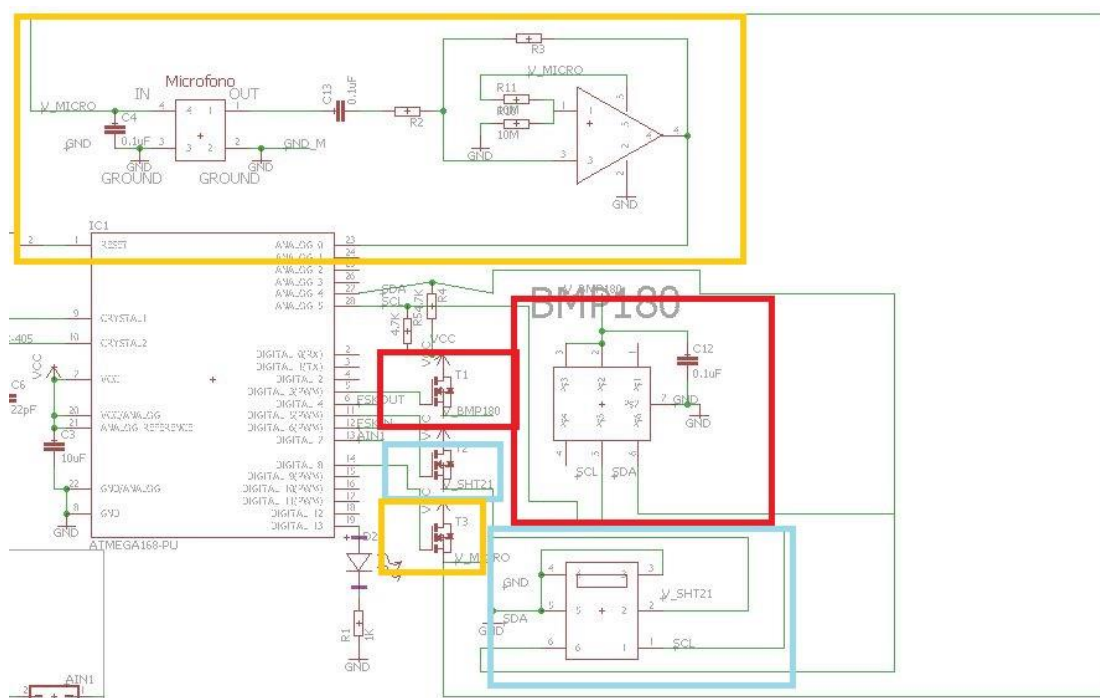


Ilustración 32. Esquemático de sensores

En la imagen se pueden apreciar 3 zonas, una por cada sensor que se ha elegido. La zona marcada en azul es la destinada al sensor de humedad SHT21 y, junto con el sensor de presión BMP180, que es la zona recuadrada en rojo, son los dos sensores digitales de los que consta el proyecto. Ambos sensores poseen la capacidad de comunicarse tanto por I<sup>2</sup>C como por SPI, que serán explicados en apartados posteriores.

Por otro lado, y de manera independiente, se tiene el circuito dedicado al sensor de presión sonora con su amplificador operacional.

### 2.2.2.1. I<sup>2</sup>C.

Tanto el I<sup>2</sup>C como el SPI son interfaces de comunicación de las que cuelgan un *master* (controlador) y uno o varios esclavos (*slaves*, dispositivos controlados). El *master* inicia la comunicación con uno o varios *slaves* a través de un identificador único para cada uno. Se ha optado por la comunicación I<sup>2</sup>C porque es una comunicación más sencilla que la SPI y sólo requiere de un canal para el reloj (SCL) y un canal para los datos (SDA).

A continuación se detallan algunas principales características del bus I<sup>2</sup>C:

- Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante firmware, disponiendo de forma permanente de una relación *master/slave* entre el microcontrolador, que suele ser el master y los dispositivos conectados.
- El bus permite la conexión de varios *masters* ya que incluye un detector de colisiones en la máquina de estados del master.
- Los datos y direcciones se transmiten con palabras de 8 bits.
- Requiere de resistencias de *pull-up*, tanto para el canal SDA como para el canal CSL.



Antes de que se establezca un intercambio de datos entre el *master* y los *slaves*, el *master* debe informar el comienzo de la comunicación (condición de Start): la línea SDA descende a cero mientras SCL permanece en nivel alto. A partir de este momento comienza la transferencia de datos. Una vez finalizada la comunicación se debe informar de esta situación (condición de Stop): la línea SDA pasa a nivel alto mientras SCL permanece en estado alto (Ilustración 33).

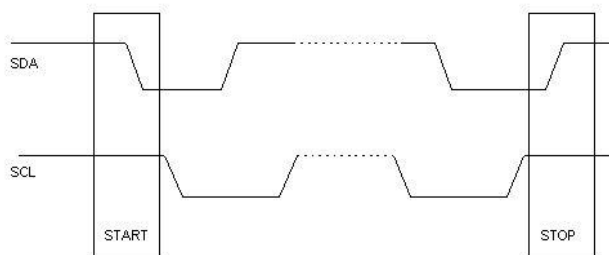


Ilustración 33. Comunicación I2C

El *master* genera la condición de Start. Cada palabra puesta en el bus SDA debe tener 8 bits, la primera palabra transferida contiene la dirección del *slave* seleccionado. Luego, el *master* lee el estado de la línea SDA, si vale 0 (impuesto por el esclavo), el proceso de transferencia continúa, si por el contrario su valor es 1, indica que el circuito direccionado no valida la comunicación, entonces, el *Master* genera un bit de stop para liberar el bus I<sup>2</sup>C. Este acuse de recibo se denomina ACK y es una parte importante del protocolo I<sup>2</sup>C. Al final de la transmisión, el *Master* genera la condición de Stop y libera el bus I<sup>2</sup>C, las líneas SDA y SCL pasan a estado alto.

En el caso de este proyecto, tanto el sensor BMP180 como el SHT21, irán conectados al bus I<sup>2</sup>C en forma de *slaves*, mientras que el microcontrolador será el que haga las funciones de master, los pines que se encargan de la comunicación I<sup>2</sup>C como se puede ver en la imagen son el pin 27 del ATmega328 para la línea de SDA, y el pin 28 para la línea del SCL.

Por último, las alimentación no irá conectada directamente a VCC, sino que irán conectadas a unos transistores de corte PMOS, cada sensor a uno. Los sensores a su vez estarán gestionados por los puertos del ATmega328, el pin 11 en el caso del SHT21 y el pin 5 para el BMP180. Gestionándolos de esta manera, se consigue un ahorro de energía considerable, evitando tener que poner en modo *low power*, por muy poco que consumieran los sensores, si se les corta la alimentación, la corriente consumida pasa a ser nula.

#### 2.2.2.2. Sensor de nivel de presión sonora.

Este sensor a diferencia de los anteriores es un sensor analógico, al que después se le debe hacer un tratamiento a la señal que nos es proporcionada. Por este motivo se explica de manera independiente. El circuito destinado al sensor de presión sonora es el recuadrado en naranja.



La señal generada por el micrófono es amplificada. Esta señal debe ser conectada mediante un offset a la entrada del amplificador, de esta forma se consigue que, en la salida, la señal amplificada no reciba valores negativos, así se evita que la misma alcance el valor 0. Además se ha añadido un filtro paso banda centrada en 4kHz, que es la banda de frecuencias que se quiere captar.

El sensor es alimentado de la misma manera que los sensores anteriores, a través de un transistor de corte. En este caso, el transistor de corte estará controlado por el microcontrolador a través del pin 14.

### 2.2.3. Implementación y pruebas iniciales.

Una vez seleccionados todos los sensores necesarios para nuestra placa y vistos los circuitos necesarios para estos, se procede a diseñar la placa que albergará los sensores y demás componentes hardware.

En este apartado se explicaran diversas posibilidades que se han estudiado para tratar de alimentar la estación meteorológica.

Como complemento al circuito *harvester*, del cual que se hablará en el apartado 2.3, se quiso implementar un sistema con un convertidor continua-continua como método para regular la señal y tener una tensión de alimentación estable. El componente seleccionado fue el TPS62052, siguiendo el siguiente esquema.

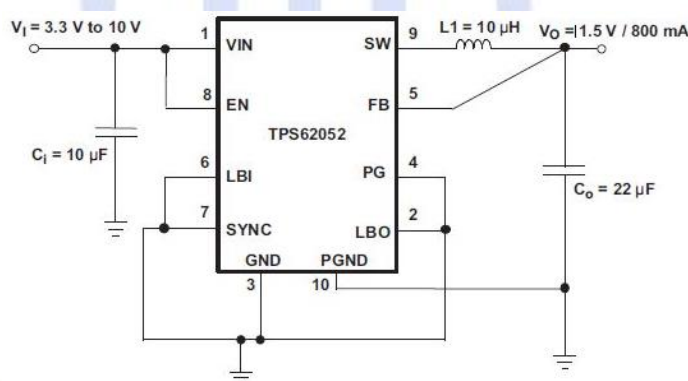
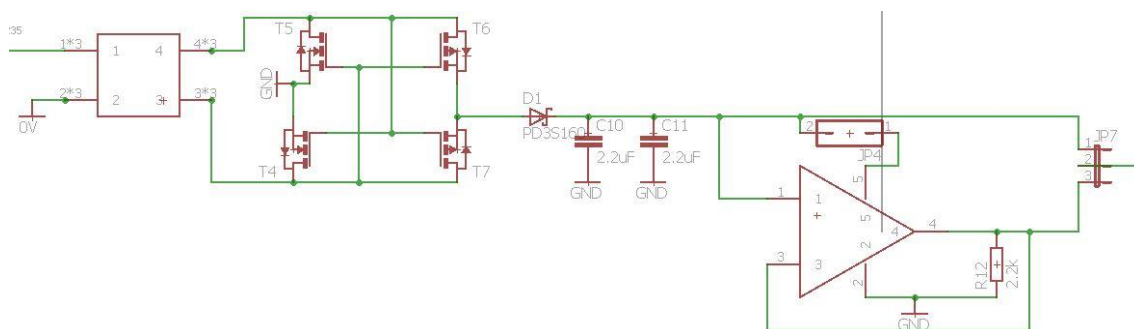


Ilustración 34. Esquemático convertidor

Un método para tratar de adaptar las impedancias entre *harvester* y teléfono, fue tratar de introducir un amplificador operacional, trabajando en modo de buffer o seguidor de tensión, alimentándose este de la tensión generada por el circuito *harvester*.



**Ilustración 35. Esquemático circuito *harvester* con amplificador operacional**

Como se puede ver en la Ilustración 35. Esquemático circuito *harvester* con amplificador operacional, para realizar las distintas pruebas, todos estos circuitos fueron integrados en la misma placa, pudiendo perfectamente conectar y desconectar los componentes a través de *jumpers*, por los que se seleccionan una parte del circuito u otra. De esta forma, se pueden probar hasta tres formas de alimentar el microcontrolador:

- Alimentación mediante *harvester* pasando por el amplificador operacional en modo seguidor de tensión.
- Alimentación directa a través del *harvester*.
- Alimentación a través de una pila.

Otra parte para finalizar el diseño de toda la placa es el circuito de adaptación de impedancias entre el teléfono y el microcontrolador para que la comunicación se dé satisfactoriamente. El diseño de ese circuito se explicará en detalle en apartados posteriores.

Una vez diseñado todos estos puntos el esquemático final queda de acuerdo a lo mostrado en el *Anexo 1.1*.

### 2.2.3.1 Pruebas hardware.

Tras haber fabricado la placa y soldado todos los componentes se procedió a realizar las distintas pruebas de funcionamiento. Las pruebas realizadas sobre el sistema de comunicación se explicaran en apartados posteriores.

Se realizaron pruebas para comprobar el buen acople entre el *harvester* y microcontrolador, con el amplificador operacional en modo seguidor de tensión, alimentando este con la propia tensión saliente del *harvester*. También se realizaron pruebas tratando de conectar directamente el *harvester*. En ambos casos el resultado obtenido fue negativo, en ningún caso se consiguió alimentar el microcontrolador y hacerlo funcionar. Esto es debido principalmente a la baja eficiencia que tiene el circuito *harvester* en comparación con el estudio de referencia de la University of Michigan [4] como se podrá ver en el apartado 2.3. Por falta de tiempo no se ha podido hacer un estudio más minucioso, sobre las causas. Sin embargo se deja instalado en la placa para futuras pruebas que se puedan realizar.

Por último con respecto al tema de la alimentación se probó a alimentar el circuito con la pila, obteniendo un resultado positivo. Se puede visualizar el video en el [Anexo 3.1](#).

Con respecto a los sensores instalados en la placa, se realizaron pruebas de manera individual con cada sensor permitiendo la comprobación de su funcionalidad. Dichas pruebas fueron realizadas previamente a establecer la comunicación entre el microcontrolador y el teléfono, por lo que era necesario hacer uso de alguna plataforma para poder visualizar el funcionamiento. Para ello, desde la placa Arduino se estableció conexión con la interfaz

disponible en el IDE permitiendo la visualización de los resultados obtenidos a través del monitor serial.

Por un lado, se probó el sensor de presión para evaluar su funcionamiento con resultado positivo se puede visualizar el video en el [Anexo 3.2](#).

### 2.2.3.1.1 Sensor de presión sonora.

A continuación se explican las pruebas que se hicieron con el único sensor analógico que se integra en el sistema, el sensor de presión sonora. El objetivo que se persigue es poder captar los niveles de presión sonora. Estos niveles harán referencia a los ruidos externos al paciente, tales como coches en marcha, electrodomésticos, etc. Lo que se pretende es captar el ruido durante un periodo de tiempo de 10 segundos, lo que es conocido como integración lenta. De esta manera, los ruidos puntuales se verán reducidos y cobrarán importancia ruidos de larga duración como la música ambiente.

Primeramente se realizaron pruebas para comprobar el correcto funcionamiento del sensor. Estas pruebas consistieron de medir con un osciloscopio y comprobar que la señal eléctrica registrada, correspondía con el ruido. Esto se comprobó emitiendo un audio de un 1kHz de frecuencia, y posteriormente viéndolo en el osciloscopio.

Posteriormente esta señal eléctrica registrada tiene que ser adquirida por el microcontrolador. En este punto se realizaron pruebas para medir la frecuencia de muestreo que tiene el ADC del microcontrolador, estas pruebas revelaron que la frecuencia de muestreo es muy cercana a los 4 kHz. Esto supone un problema ya que se quiere poder muestrear una onda que tenga un ancho de banda de 4kHz, para poder muestrearla, según el teorema de Nyquist debemos muestrearla a por lo menos el doble de la frecuencia.

El método para acelerar la adquisición de datos del ADC del microcontrolador<sup>13</sup> consiste en configurar los bits del ADC en el código del firmware, quitándole el *prescaler* preconfigurado en el mismo. El *prescaler* es un valor por el cual se divide la frecuencia, este valor es 16 inicialmente. De esta forma se puede llegar a muestrear a frecuencias más altas, por lo que se lleva a cabo una nueva comprobación.

La prueba consiste en muestrear dos ondas, una de 1 kHz y otra de 8 kHz, usando el código recogido en la función "AnalogRead ()". Si dicho método consigue realizar las mediciones deseadas se obtendrá margen suficiente como para muestrear una onda y analizar los datos obtenidos con una frecuencia de, al menos, 4 kHz.

Los microcontroladores ATmega328, según la hoja de especificaciones del mismo<sup>14</sup> configuran el *prescaler* del ADC con tres bits de acuerdo con la siguiente figura (Ilustración 36).

<sup>13</sup> <http://booleanbite.com/web/adquisicion-de-datos-con-arduino-i-tiempo-de-muestreo-y-resolucion/> Última vez consultada: 15/05/2016

<sup>14</sup> [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf) Última vez consultada: 15/05/2016

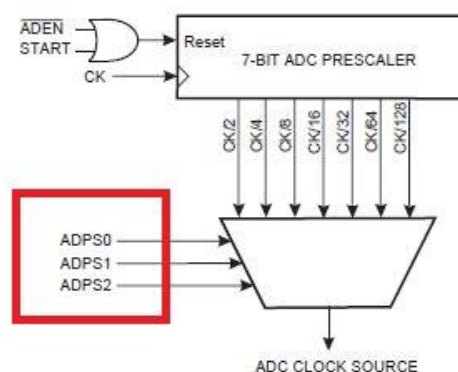


Ilustración 36. Prescaler en ATmega328<sup>29</sup>

Para realizar estos cambios en el *prescaler* del ADC. Se configura el ADC con sentencias de precompilador para conseguir una frecuencia de 64kHz. Se establece esta frecuencia para tener margen de introducir instrucciones para manejar los datos obtenidos por el ADC.

Una vez aumentada la frecuencia de muestreo. Se procedió a muestrear una onda de 1KHz y otra de 8 KHz para comprobar el correcto funcionamiento del ADC. Los resultados son los siguientes (Ilustración 37).

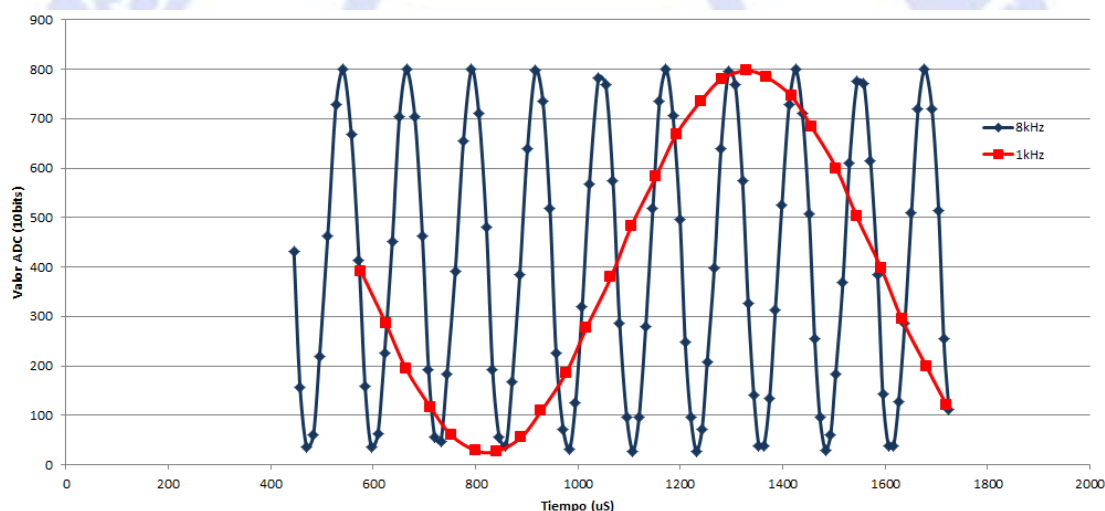


Ilustración 37. Comparativa adquisición dos ondas

Como se puede ver en la Ilustración 37, se consigue representar perfectamente las ondas senoidales. Como se ha comentado anteriormente esto facilita poder incluir instrucciones para tratar los datos obtenidos de la señal de ruido muestreando una señal de 4Khz.

A partir de este momento se comienza a incluir las instrucciones necesarias para la integración de los niveles de presión sonora durante un periodo de tiempo de 10 segundos.

Primero se obtiene el dato, dentro de una estructura repetitiva llamada *for* que se incrementa:

- Se elimina la componente de continua de 1.5V.
- Conversión de la muestra a voltios, con un fondo de escala igual a 3.3V.
- Como se están calculando potencias, se eleva el valor al cuadrado.
- Por último se añade el valor tratado a la variable que va acumulando todos los valores de potencia.

Una vez que termina la ejecución de la sentencia *for*, se trabaja con el valor de potencia obtenido en este ciclo durante los 10 segundos:



- Se divide el valor obtenido por el valor de la ganancia final al cuadrado que se tiene después de pasar la señal por el micrófono<sup>15</sup>. Aplicamos la siguiente fórmula para convertir el dato en dB SPL. dB SPL es el nivel de presión sonora, que determina la intensidad del sonido que genera una presión sonora (es decir, del sonido que alcanza a una persona en un momento dado), se mide en decibelios (dB) y varía entre 0 dB umbral de audición y 120 dB umbral de dolor.

$$SPL = 10 * \log \frac{\text{Valor medido}}{\text{Referencia}^2}$$

El valor de referencia es el valor obtenido directamente de la hoja de características del sensor, que es -42 dBV. Este valor es la sensibilidad del micrófono. El valor medido no se ha elevado al cuadrado porque ya se ha realizado durante el proceso de medición.

## 2.3. Diseño del sistema de alimentación: *energy harvester*.

Una de las primeras pruebas realizadas fue la investigación respecto a la corriente soportado por el teléfono para poder alimentar la estación meteorológica. Como se está buscando un sistema cómodo para el paciente, en principio se comienza descartando la alimentación USB, ya que va a ser un sistema que lleve a cabo mediciones durante las 24 horas del día. Este enfoque entra en conflicto con la idea de que el paciente pueda cargar su teléfono, por lo que se buscan alternativas. Aunque para solucionarlo se podría añadir un *splitter* con el cual se pudiera elegir entre cargar el teléfono o usar la estación meteorológica, esto supondría un aumento en la complejidad del sistema y en el coste del mismo. Por esa razón se ha estudiado la alimentación mediante el conector *jack* del propio teléfono. La alimentación, que se desea obtener a través de una señal de audio proveniente del teléfono, servirá para hacer funcionar el sistema. Se pretende lograr el funcionamiento de todos los componentes en modo *low-power*, por lo que se va a requerir una corriente de alimentación del orden de miliamperios. En cuanto al voltaje, el sistema requiere que el ATmega328 pueda encenderse, por lo que se requieren entre dos y tres voltios.

Antes de estudiar el sistema colector de energía es necesario precisar cuanta energía es capaz de generar el teléfono móvil.

### 2.3.1. Obtención de energía del teléfono.

En un trabajo anterior de Kuo de la University of Michigan llamado "*Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface*" [4], se realiza un experimento cuyo objetivo es crear una interfaz analógica con la que el teléfono pueda alimentar a dispositivos externos y comunicarse con ellos a través de su puerto *jack*. Para ello, se diseña un dispositivo el cual es capaz de, partiendo de una onda senoidal proporcionada por el teléfono, rectificarla y proporcionar una tensión continua. En este proyecto se va a tratar de imitar este sistema.

Primeramente para poder estimar la corriente que se puede obtener a través del *jack* del teléfono móvil se realizó el siguiente experimento: conectando el canal de audio del conector *jack* a una resistencia de carga variable, se calculó la corriente y potencia que ofrece el teléfono a través de la medida de la tensión caída en la resistencia de carga mientras se emite

<sup>15</sup> [https://es.wikipedia.org/wiki/Nivel\\_de\\_presi%C3%B3n\\_sonora](https://es.wikipedia.org/wiki/Nivel_de_presi%C3%B3n_sonora) Última vez consultada: 15/05/2016



una señal de audio. En particular, se utilizó la aplicación *Signal Generator* para generar una onda senoidal, y se usaron distintas resistencias de carga.

En estos experimentos surgieron dos complicaciones. Uno de los problemas es que en un principio el teléfono, no detectaba el conector *jack*, la solución consistió en conectar el *jack* a tierra para que el teléfono detectara un conector de 3 canales. El segundo problema es que el teléfono móvil tiene por defecto un nivel de volumen que no es el máximo posible. Por tanto, es necesario subirlo, pero una vez transcurrido cierto tiempo, el sistema operativo se encarga de establecer de nuevo el valor estándar del volumen. Por lo que más adelante se observará como mediante Android se ha establecido un volumen en la aplicación y se han bloqueado las teclas del teléfono para que no se pueda bajar el volumen. Además, se observó que reproduciendo, a través del teléfono, una onda de audio generada en formato WAV (mediante la aplicación Audacity), se obtenía una mayor energía que con las señales dadas por *Signal Generator*.

A continuación (Ilustración 38) se muestran los valores obtenidos para la potencia y la corriente, según va cambiando la resistencia, esta medición se realiza midiendo la tensión eficaz de la onda senoidal generada por el teléfono en la resistencia de carga.

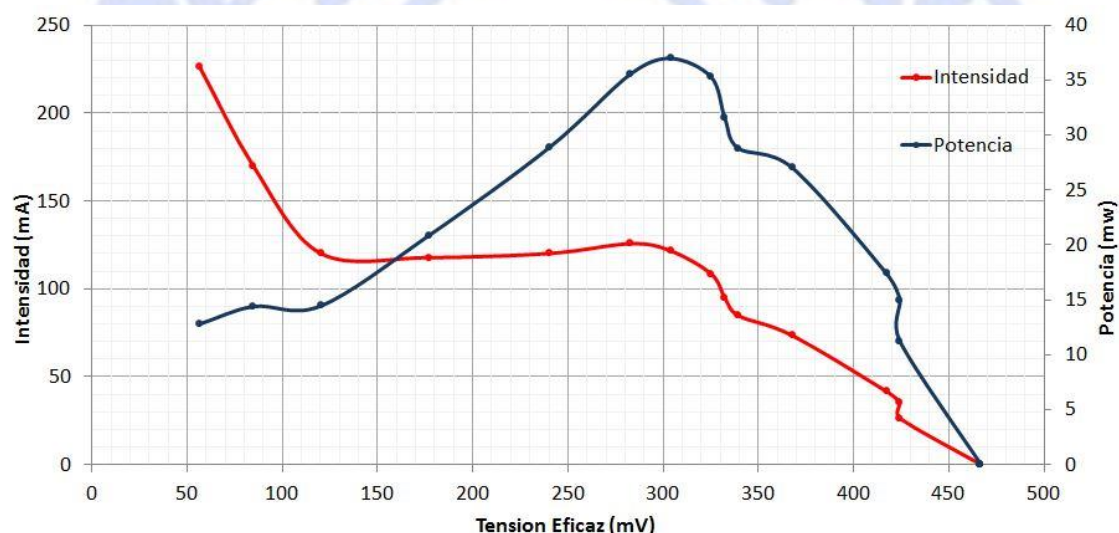


Ilustración 38. Potencia obtenida en Motorola Moto G

Como se puede ver, su pico máximo alcanza algo más de 37 mW con una tensión eficaz de aproximadamente 300 mV.

A modo de prueba, se comparó el resultado con el obtenido para otro terminal móvil: el teléfono BQ Aquaris<sup>16</sup>. La curva de energía proporcionada por este dispositivo se muestra en la siguiente ilustración (Ilustración 39). En esta gráfica se aprecia como la potencia obtenida está por debajo de la potencia resultante de la gráfica anterior. En la gráfica anterior se ve que la potencia generada es 37 mW, mientras que la segunda gráfica arroja que la potencia máxima que otorga el BQ Aquaris son 1.57 mW, esto es debido a que en la primera gráfica, para llegar a los 300 mV donde se alcanza la potencia máxima, la resistencia de carga es 2.5  $\Omega$  mientras que en la segunda gráfica para alcanzar la potencia máxima es necesario aplicar una resistencia de carga de 600  $\Omega$ . Esta potencia es insuficiente para alimentar algún accesorio externo.

<sup>16</sup> <https://www.bq.com/es/aquaris-a4-5>

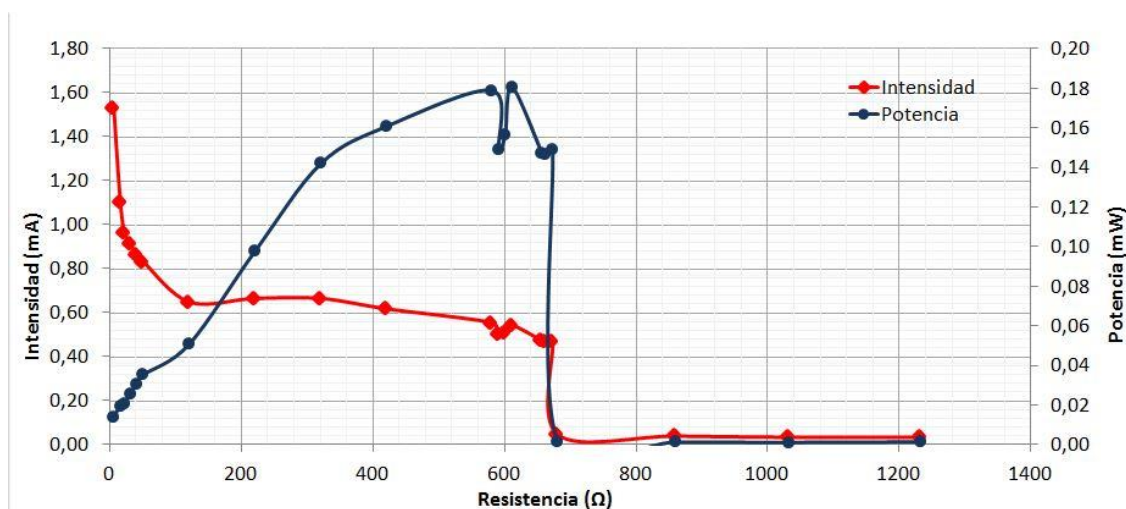


Ilustración 39. Potencia obtenida BQ Aquaris

Esta cuestión puede ser una de las razones por las que accesorios comerciales como Windoo, se limitan a ofrecer su producto a un listado de modelos de Smartphones más vendidos, los cuales han podido realizar las pruebas de alimentación y funcionamiento.

### 2.3.2. Diseño inicial del sistema de alimentación.

En el trabajo “*Hijacking Power and Bandwidth from the Mobile Phone’s Audio Interface*” [4] se destina un canal de audio para generar la tensión de alimentación, utilizando el circuito mostrado en el [Anexo 1.6](#). Cabe destacar que el puerto *jack* no es detectado con la misma eficacia según en qué dispositivo móvil, en unos dispositivos móviles lo detecta con toda facilidad, ya que depende del SO instalado en el teléfono, por lo que es necesario para este experimento conectar el canal del micrófono a la de tierra del circuito para su correcta detección en todos los dispositivos.

Este circuito, en [Anexo 1.6](#), se basa en un micro-transformador con una relación de transformación 1:20, cuatro transistores FET (dos tipo N y dos tipo P) que forman un puente FET consiguiendo una potencia de 15 mW. Con un diodo Schottky se consigue que la señal retorne al puente FET y a continuación, dos condensadores en paralelo para terminar de eliminar cualquier tipo de rizado.

Se pretende reproducir dicho sistema y caracterizarlo para comprobar si nos permite alimentar el sistema de monitorización de variables ambientales objeto de este Trabajo Fin de Grado.

### 2.3.3. Implementación y pruebas iniciales.

A continuación se enumeran los componentes que se han utilizado en el montaje de ese sistema.

Componente	Marca	Modelo	Cantidad
Jumper			3
Transformador	Coilcraft	LPR6235	1
Transistor-N	Zetex	ZXM61N03	2
Transistor-P	Zetex	ZXM61P03	2
Diodo Schottky		DFLS120L	1
Condensador			2

Tabla 10. Componentes utilizados en el circuito *harvester*

El modelo de micro-transformador es introducido porque permite obtener de acoplamiento por encima del 95% con un tamaño muy reducido. El resto de componentes son convencionales. Se ha diseñado una PCB con componentes de montaje superficial (SMD por sus siglas en inglés) para realizar las pruebas.

Para la creación de la placa se crearon las huellas para el transformador debido que es un modelo poco común no presente en las librerías de EAGLE. Los *layouts* quedan como se muestra en [Anexos 1.2 y 1.3](#).

Las impedancias entre la entrada del micro-transformador y la salida del teléfono deben ser igualadas, de esta forma se evitan reflexiones por desacoplos de impedancia. Para hacer esto se debe tener en cuenta los valores de la inductancia del primario del transformador, y la impedancia de salida. Aplicando la fórmula de la impedancia de un inductor, nos permitirá encontrar la frecuencia de la señal de excitación del *harvester*.

$$X_L = j\omega L = j2\pi fL$$

La impedancia de salida de los del *jack* de audio es muy baja, está alrededor de los tres ohmios, y es el valor donde se obtiene la mayor potencia, como se ha podido ver en el apartado 2.2.1. Según la hoja de características del [Anexo 2.1](#), la inductancia de este modelo son 25  $\mu$ H. Por lo que si igualamos las impedancias a ambos lados, tenemos:

$$f = \frac{3 \Omega}{2\pi * 25\mu H} = 19098.6 \text{ Hz}$$

Una vez que conocida la frecuencia a la que debe estar la onda generada se procede a caracterizar la curva de energía del dispositivo de la misma manera que se hizo con el teléfono móvil (variando una carga a la salida y excitando el circuito con una señal sinusoidal de 19 kHz. Los resultados son los siguientes.

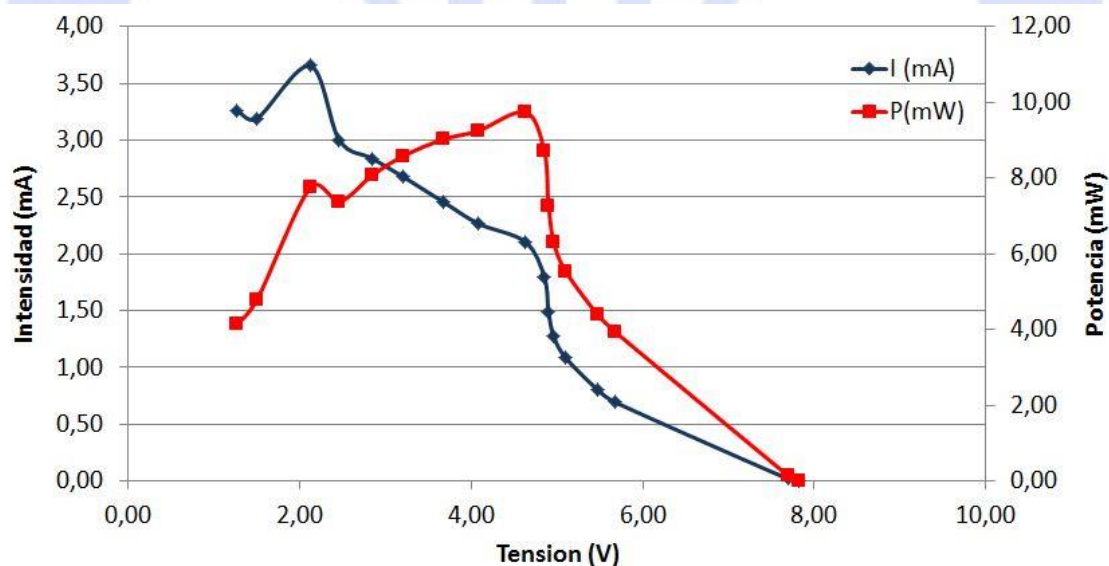


Ilustración 40. Potencia *harvester*

En esta gráfica (Ilustración 40) se aprecia que la potencia máxima generada a 19kHz por el *harvesting* se sitúa en torno a 9.7 mW con 4.63 V a la salida. Esta medida no corresponde con lo esperado, ya que en el trabajo anterior [4] la eficiencia de conversión es del 66%, y en el experimento se obtiene un 26%.

Las causas concretas de estas pérdidas de eficacia se desconocen. Una de las posibles causas puede estar en el tamaño de la placa de pruebas que se ha realizado, siendo este mayor que

en el experimento del trabajo de la University of Michigan, unido a la posibilidad de un tamaño de pista más pequeño y a la no simetría en la colocación de los componentes del puente FET puede haber afectado, sin embargo, las pérdidas parecen excesivas para ser únicamente explicado por estas causas.

A raíz de esto se trató de realizar otras pruebas sustituyendo el diodo Schottky por otros componentes ya que se sospechó que el diodo podía no estar bloqueando correctamente la señal. Se probó sustituyéndolo por un diodo común (1N4007), con este mismo diodo además de un Amplificador Operacional con la configuración invertida, para que si la vea una impedancia muy alta, y por lo tanto bloquee la señal y no retorne al puente FET. Por último, se probó con otro modelo de diodo. Los resultados se muestran en la siguiente tabla.

Componente	Pmax (mW)	Voltaje (V)	Intensidad(mA)
1N4007	4,26	2,26	1,88
1N4007+OAMP	4,51	2,6	1,73
Invertido			
1N4148	3,35	2,24	1,49
Schottky	9,74	4,63	2,01

Tabla 11. Medidas obtenidas en las pruebas de implementación del *energy harvesting*

Como se puede apreciar todas estas pruebas revelan una pérdida mucho mayor que con un diodo Schottky. Los resultados finales arrojan una potencia máxima de 9.74 mW, a una tensión de 4.63 V con una corriente de 2.01 mA. Se podría llegar a obtener más corriente sacrificando la potencia máxima suministrada, en función de cuanta tensión fuera necesario suministrar, como se puede ver en la Ilustración 40.

Por último, se hace una comparativa entre las medidas realizadas con las señales generadas por la aplicación *Signal Generator*, y las realizadas con archivos de audio con el formato WAV. Para ello se ha utilizado la caracterización de la energía proporcionada por el *harvester* anterior, y se ha realizado la misma caracterización con ondas creadas anteriormente con Audacity. El resultado es el que se muestra en la siguiente gráfica.

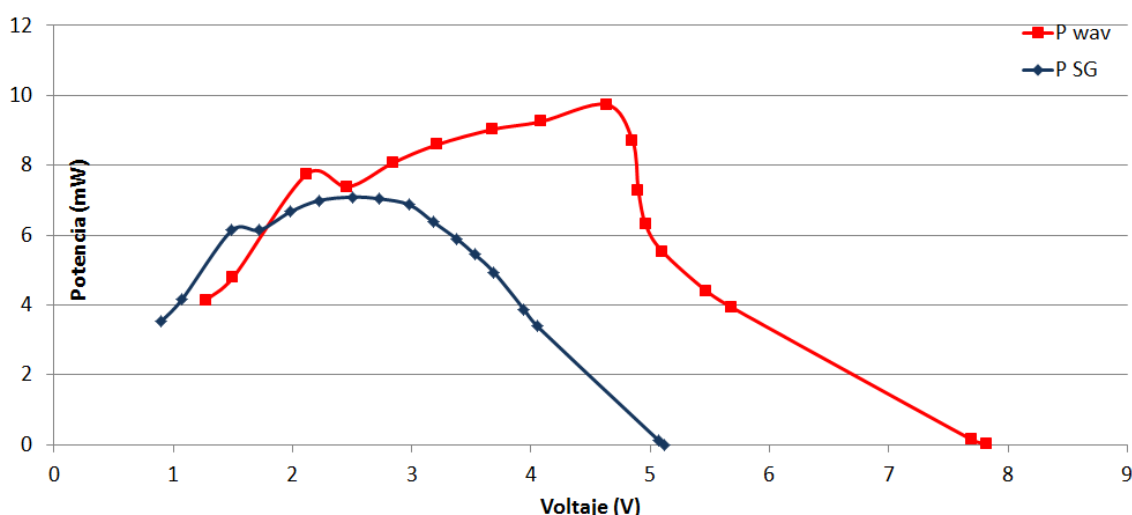


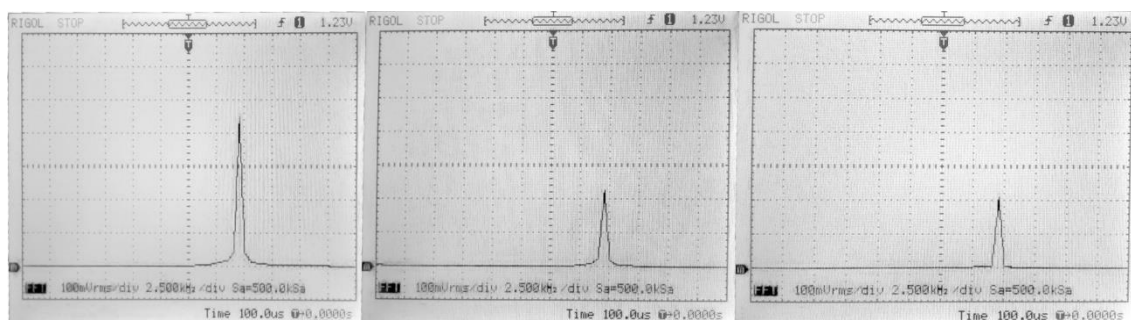
Ilustración 41. Comparativa potencias entregadas

Como se aprecia en la Ilustración 41, la energía obtenida es algo más de la mitad de la onda generada anteriormente. Esto se debe a que la cantidad de armónicos que hay en una



señal WAV aporta energía que por otro lado no está presente en los tonos puros generados por la aplicación o cuando se usa una señal de audio comprimida tipo .MP3.

En la siguiente imagen se compara tres señales senoidales, una reproducida en la aplicación *Signal Generator*, y otras dos que tienen un formato MP3 y WAV. Se ha analizado el espectro de distorsión armónica de cada una de las señales.



**Ilustración 42. 1. Onda formato WAV (Izquierda). 2. Onda generada con Signal Generator (centro). 3. Onda formato MP3 (derecha)**

En la imagen de la derecha, se muestra una señal recreada con la aplicación *Signal Generator* pudiendo apreciar su semejanza a la señal que se observa en la imagen del centro, que tiene un archivo en formato MP3 (con compresión logarítmica en amplitud y frecuencia). Por el contrario, en la imagen izquierda se aprecia el resultado obtenido a partir de la señal creada con un archivo formato WAV, en dicha imagen se puede ver la frecuencia deseada con mayor amplitud, por esta razón es el formato elegido para reproducir el sonido.

### 2.3.4. Conclusiones y sistema final.

En los apartados anteriores, se ha explicado la potencia obtenida en el dispositivo tiene un valor de 9.74 mW. Esto se traduce en aproximadamente 2mA, con una tensión continua de alrededor de los 4.6V. Esta potencia energía es bastante limitada para nuestra aplicación ya que esta demanda más potencia.

A los problemas encontrados durante las pruebas realizadas se le añadió una dificultad aún mayor: la adaptación de las impedancias entre el *harvester* y la alimentación del sistema para no perder energía. Este adaptador debería poder ser un adaptador de impedancias pasivo, la energía obtenida del *harvester* es muy limitada y no conviene reducirla más. Por otro lado, se ha buscado otros medios para aislar las impedancias como, por ejemplo, un amplificador operacional en modo seguidor o un convertidor continua-continua.

Dadas todas las dificultades encontradas, así como la potencia limitada que se consigue a través del *harvester*, se ha implementado una alimentación suplementaria en forma de batería. En particular, JH752540P que tiene una capacidad 500 mAh, una tensión de alimentación 3.7V.

Por otro lado, en la aplicación Android se ha dejado implementado que al habilitar el micrófono para que empiece la comunicación, el teléfono emita una onda senoidal por el canal de audio que alimenta el circuito *harvester*. Para futuras líneas de trabajo en las que se mejore la eficiencia del *harvester* y poder alimentar el microcontrolador.



## 2.4. Adaptador de impedancias.

Como se ha comentado en otros apartados, una parte fundamental de este trabajo es conseguir una buena comunicación entre la estación meteorológica y el microcontrolador para que se almacene la información obtenida. El canal por el que se va a realizar esa comunicación es el conector *jack* del teléfono. Esta manera es la forma elegida ya que los últimos modelos del puerto *jack* poseen 4 vías, comúnmente dos son utilizados para la reproducción de música, un tercero para el micrófono en los accesorios de manos libres del teléfono, y el 4 para comunicar la tierra de ambos dispositivos. Una imagen del conector *jack* se puede ver en la siguiente figura (Ilustración 43).

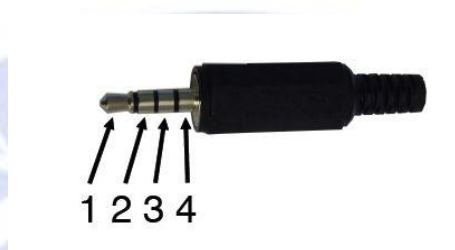


Ilustración 43. Conector *jack* de audio TRRS

Para la aplicación esas características son muy positivas, ya que por un lado se dispone de un canal de audio para tratar de alimentar la estación meteorológica emitiendo una onda senoidal desde el teléfono para que luego el *harvester* se encargue de rectificarla y convertirla en continua, mientras que con el otro canal se le puede mandar comandos al microcontrolador. En este proyecto este canal se utilizará para despertar el microcontrolador como se puede ver en el [Anexo 1.4](#). El canal del micrófono se utiliza para mandar los datos recogidos por los sensores al teléfono.

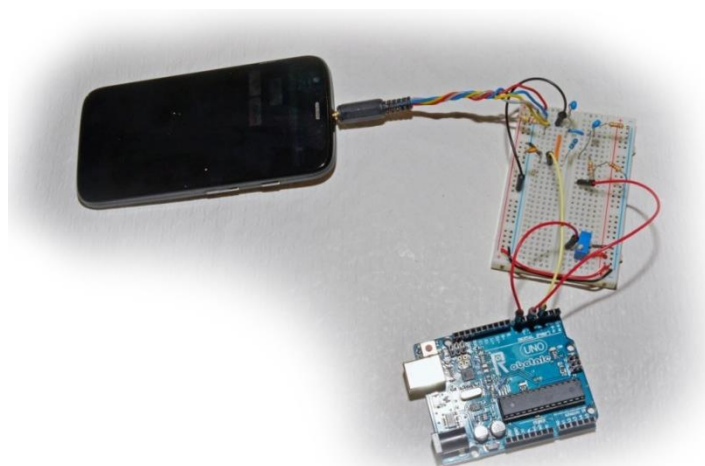
Una vez conocidos los pines a utilizar en el microcontrolador para realizar la comunicación se implementa el circuito hardware. Para realizar este circuito hardware nos basamos en el esquema de un circuito comercial. Nos hemos basado en este circuito porque es un módulo<sup>17</sup> utilizado en algunas aplicaciones con el mismo principio, comunicarse con teléfonos inteligentes.

En el esquemático del [Anexo 1.4](#) se pueden apreciar que al teléfono tiene 3 canales, que son los dos canales de comunicación, nombrados en la figura como FSKIN y FSKOUT además de la conexión de tierra. Sin embargo, de cara al microcontrolador se aprecian 5 conexiones, que son: alimentación, tierra, FSKIN, FSKOUT y una última conexión llamada en el [Anexo 1.4](#) como AIN1. Esta última conexión es muy importante ya que la referencia que se le tiene que dar a microcontrolador según la hoja de características<sup>18</sup> para que haga las veces de trigger. Los valores que sean superiores al valor de AIN1 serán considerados como un “1” lógico y los valores que sean inferiores serán considerados como “0” lógico.

Primero se analiza el funcionamiento del circuito. Para comenzar, se implementa el circuito del [Anexo 1.4](#) en una placa de pruebas para analizar la señal en cada punto del circuito. También se sueldan cables al conector *jack* para poder conectar la placa al teléfono (Ilustración 44).

<sup>17</sup> <http://trac.switch-science.com/wiki/ARMS22-SOFTMODEM-HOWTO> Última vez consultada: 15/05/2016

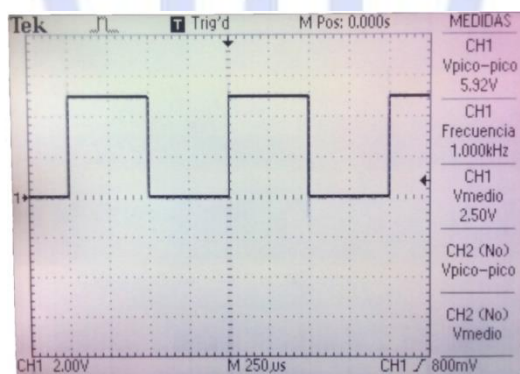
<sup>18</sup> [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf) Última vez consultada: 15/05/2016



**Ilustración 44. Implementación placa de pruebas del adaptador de impedancia**

Con esta implementación se procede a probar la comunicación. Se trata de mandar mensajes individuales controlados, desde el microcontrolador al teléfono (se usa Arduino para estas pruebas). Los valores de tensión para que la comunicación con el móvil hasta el microcontrolador fuese correcta es de 1.84V en el caso de alimentarlo con 5V y 1.01V si se alimenta con 3.3V (valores empíricos). A continuación, se puede ver un ejemplo de osciloscopio del envío de datos desde el teléfono ( Ilustración 45 y Ilustración 46)

Una vez configurado el canal en el sentido teléfono-microcontrolador, se realizan las pruebas en el sentido contrario.



**Ilustración 45. Señal emitida por el microcontrolador**

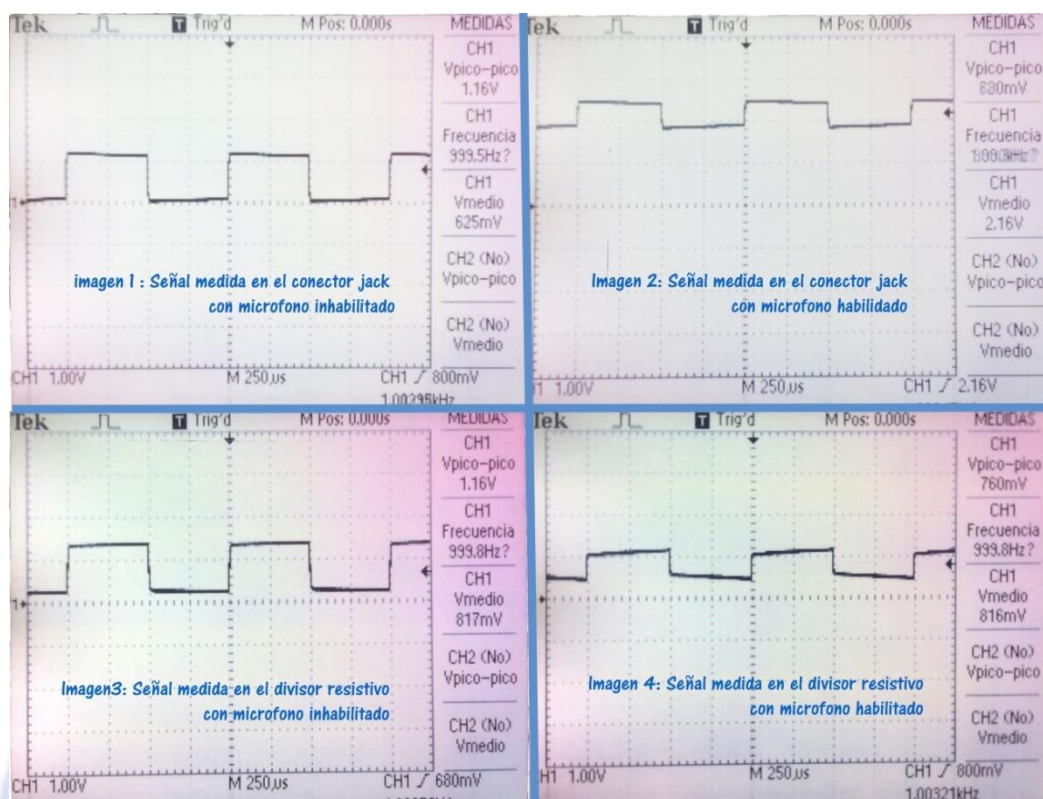


Ilustración 46. Comparativa señales

En la Ilustración 45 se puede ver la señal saliente del microcontrolador, una señal cuadrada de 5 voltios pico-pico centrada en 2.5V, esta es la señal que el microcontrolador envía. En la imagen siguiente (Ilustración 46) se pueden ver cuatro capturas de osciloscopio.

Como se puede ver en la Ilustración 46, en el momento en que se enciende la aplicación para poder iniciar la comunicación, la onda se deforma en gran medida. Además aparece un offset, proporcionado por el teléfono, por estos motivos se cambia el circuito de este canal del adaptador de impedancias.

Modificando los valores de las dos resistencias que se encuentran en este canal de comunicación, se observó que realizaba una mejor adaptación, obteniendo una forma de onda mucho más cuadrada y con mayor amplitud, como se puede ver en la siguiente figura (Ilustración 47), donde se muestra la señal cuando está habilitado el micrófono a la entrada del teléfono.

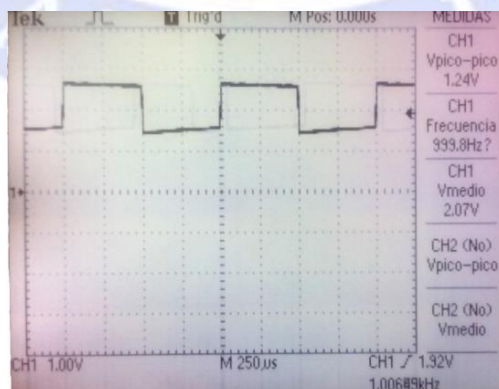


Ilustración 47. Señal con las resistencias invertidas.

Continuando con esta tendencia, se realiza un cambio en la colocación de los componentes, llegando a dejar el circuito como se muestra en el [Anexo 1.5](#).

Cambiando la disposición de la resistencia que estaba ubicada a la salida del microcontrolador, se mejora la calidad de la señal a la entrada del teléfono como se puede ver a continuación.

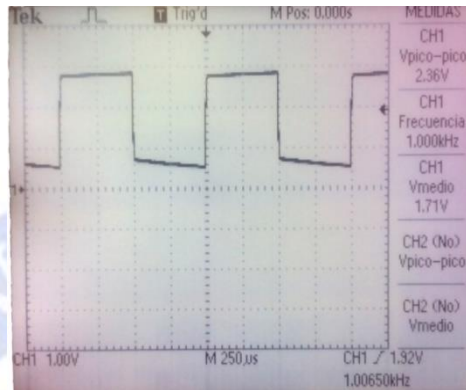


Ilustración 48. Señal final adaptador de impedancias







## Capítulo 3. Diseño del sistema software.

En este capítulo se procederá a explicar cuáles son las acciones que se han realizado a nivel de software.

Por un lado se ha utilizado Android para implementar la comunicación por parte del teléfono, mostrar por pantalla los valores de los sensores en la interfaz gráfica y gestionar el sensor de luz del Smartphone. Por otro lado para cargar el firmware en el microcontrolador de la estación meteorológica, se ha trabajado sobre la plataforma Arduino, y para el propósito de la comunicación con el teléfono por parte del microcontrolador se ha utilizado una librería creada para Arduino desde la plataforma GitHub<sup>19</sup>.

### 3.1. Sensor de Luminosidad.

La última variable ambiental que nos queda por medir es la luminosidad del ambiente, para ello se decidió realizar la medición con el sensor de luminosidad que poseen algunos teléfonos. Estos sensores suelen ubicarse en la parte frontal del teléfono, junto con un sensor de proximidad. El sensor de luminosidad se suele utilizar para regular el brillo de la pantalla en función de la luz ambiente.

Después de comprobar que el modelo de teléfono a utilizar, posee el sensor de luminosidad se procedió a realizar una pequeña aplicación que se encargase de leer ese sensor y mostrarlo por pantalla.

Para ello, se tomó como referencia un TFG [5] en el cual se trabajó con los sensores del teléfono. Lo primero que se implementó es la parte del *layout* de la aplicación, en esta parte, se declaran tres elementos. Dos elementos para visualizar texto (TextView) y una barra de progreso. Estos elementos servirán para visualizar el nivel de luz que está captando el sensor, el nivel máximo que el sensor puede llegar a captar y una relación de estos dos poniéndolo de manera gráfica.

En la parte del código de la *activity* principal consta de cuatro partes.

- El primer bloque representa la parte de código empleado para inicializar las variables.
- En el segundo bloque tiene dos constructores, primero el constructor de la clase *SensorManager*, en esta clase se gestionan todos los sensores que posee el teléfono, como son acelerómetro, sensor de proximidad o sensor de luz. En este proyecto solo se utilizará este último. El segundo constructor corresponde al sensor de luz, a partir de estos dos constructores se puede utilizar el sensor de luz y realizar la medición.
- Posteriormente se inicializa un último constructor, *SensorEventListener*. En dicho constructor, estarán las funciones principales para realizar las medidas. Principalmente la función *OnSensorChanged()*, será la que se ejecutará cada vez que la medida del sensor cambia, cuando sucede esto, el valor del sensor se actualiza en pantalla.
- Una vez creado todos estos constructores se añade una estructura para localizar si se ha iniciado bien el sensor de luz. Por último, este método analiza si se tiene inicializado correctamente el sensor de luz, en caso negativo, simplemente se muestra por pantalla un mensaje como que algo ha fallado.

Finalmente el resultado es el siguiente (Ilustración 49).

---

<sup>19</sup> <https://github.com/arms22/SoftModem> Última vez consultada: 14/06/16

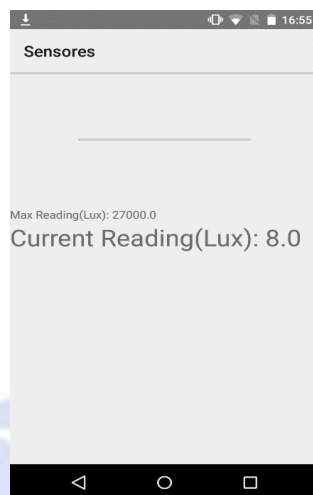


Ilustración 49. Interfaz de usuario de aplicación de luz

En la parte central de la pantalla con una letra más grande, se puede ver la intensidad lumínica medida en ese momento, encima de esa medida se encuentra el valor máximo que mide, y en la parte superior se encuentra la barra de progreso, que al estar midiendo muy poca luz al estar en oscuridad, el valor está al mínimo.

## 3.2. Sistema de comunicaciones.

En este apartado se va a explicar el sistema de comunicación entre el teléfono y la estación meteorológica. El protocolo de comunicación entre la estación meteorológica y el teléfono que se seguirá en este proyecto será el siguiente.

### 3.2.1. Codificación/decodificación microcontrolador-Android.

Inicialmente la aplicación de la que se partió<sup>20</sup> para crear la de este proyecto tenía ciertas características intrínsecas. La aplicación no podía soportar longitudes mayores a 5 bits, en decimal el valor máximo que se podría mandar es 31. Esto es debido a que en la versión original el tamaño de palabra es de 8 bits y se utilizaban tres bits para la suma de comprobación (*checksum*).

Este número de bits es insuficiente para nuestros propósitos, ya que cualquier dato que quisiésemos mandar será mayor que 31. Es más, los datos obtenidos por los sensores serán tipo *float*, de un tamaño de 16 bits, por lo que el tamaño de palabra excederá por mucho el tamaño disponible para el envío. Con el *checksum* instaurado en el mensaje los bits hábiles para mandar son 5, por lo que para mandar un dato *float*, harían falta 4 envíos. Además que el hecho de tener puesto el *checksum* puede hacer ver que el mensaje correcto no es bueno. Por estas razones se decide eliminar el *checksum* y mandar el mensaje 3 veces, y dentro de la aplicación de los tres mensajes, si un mensaje se repite al menos dos veces, significara que es correcto.

Con una comunicación de 8 bits por envío, las variables *float* continúan siendo demasiado pesadas para unirlos en un solo envío. Por este motivo, se dividirán en la parte entera y la parte decimal y una vez recibidas en la aplicación, se unirán. Esto sirve para el envío de todas las variables salvo el de la presión atmosférica. En este caso, como los valores de presión

<sup>20</sup> <http://androino.blogspot.com.es/p/project-description.html> Última vez consultada: 15/05/2016

atmosférica suelen oscilar entre 900 milibares los días de tormentas, hasta los 1100 milibares. Por esta razón, antes del envío de esta variable, se le restara un valor entero de 900 y posteriormente se mandara de la misma manera que el resto de los datos, cuando llegue a la aplicación se tendrá en cuenta.

Viendo que se va a necesitar mandar fragmentados los datos, se necesita establecer una codificación para conocer que dato se está recibiendo. La codificación consistirá en mandar primeramente un número que identifica a cada sensor, para que la aplicación sepa de cual se trata, dichos números pueden ser consultados en el [Anexo 5](#).

Se han utilizado valores altos para garantizar que en ningún momento un valor pueda tener esos valores. Valores de temperatura superiores a 50º C de temperatura, 100% de humedad relativa, 120 dB de ruido, o 200 *mbar* (1100 *mbar* -900 *mbar*) de presión atmosférica son altamente improbables de medir.

- La comunicación se realizara una vez cada 10 minutos.
- Cuando pasa ese tiempo, el teléfono manda una señal con la que despierta el microcontrolador.
- El microcontrolador, toma las medidas de las los sensores y los envía al teléfono.
  - Primero se envía un dato, que sirve para indicar el tipo de variable que se enviará posteriormente. La codificación de los datos se puede ver en el [Anexo 5](#).
  - El dato enviado se envía en dos partes, primero la parte entera del dato, y después la parte decimal. Luego en la aplicación Android se unen de nuevo.
  - Cada dato se envía tres veces. Con esto se evitan fallos de comunicación. En la aplicación Android se comparará entre las recepciones, y si al menos dos datos son iguales es guardado, ya que será un dato correcto.
- Una vez enviados todos los datos, el microcontrolador se vuelve a poner en modo sueño, y la aplicación Android deshabilita micrófono hasta que vuelva a pasar el tiempo estipulado para el nuevo envío (10 minutos).

### 3.2.2. Comunicación FSK.

Primero se implementa el código para el microcontrolador necesario para comunicarse con el teléfono.

En el proyecto se ha implementado una comunicación semi- dúplex asíncrona, el enlace entre emisor y receptor se realiza en cada palabra de código transmitido. Para que puedan comunicarse, emisor y receptor deben tener la misma velocidad de transmisión. El primer dato propio del mensaje es el bit menos significativo (LSB).

El formato del mensaje es el siguiente: se debe iniciar el mensaje con un bit de start. Posteriormente el paquete contiene ocho bits de información. Esto se ha modificado de la librería original ya que inicialmente el mensaje se mandaba con un *checksum*, pero dada la pérdida de bits que supone, se eliminó este *checksum*, para aumentar la capacidad del mensaje. La pérdida de la comprobación de la llegada correcta del mensaje, se sustituyó por un aumento el número de envíos que se realiza.

A continuación se muestran dos imágenes para comparar dos datos diferentes para identificar los distintos bits dentro de la onda (Ilustración 50).



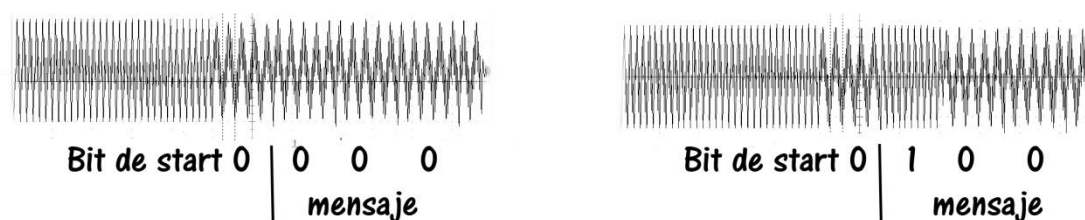


Ilustración 50. Comparación de señales FSK

En la imagen se aprecian dos señales emitidas desde el teléfono móvil hacia el microcontrolador. En la señal de la izquierda se ha emitido un 0 decimal codificado después del bit de start como 00000000, y en la de la derecha se emite un 1 decimal codificado como 00000001. Las frecuencias base de la modulación son 3150 Hz para los bits “1” y 1575 Hz para los bits “0”. De forma general en ambas señales se puede apreciar como antes de comenzar el mensaje emitido, la onda tiene una frecuencia que equivale al “1” lógico hasta que llega un “0”, esa es la señal que indica que va a comenzar a recibirse un mensaje. Para finalizar el mensaje, no se muestra en la imagen, terminará con un bit de parada que es un “1” lógico y la comunicación quedará interrumpida.

### 3.2.2.1. Firmware del microcontrolador.

Para implementar la modulación FSK en microcontrolador, existe una librería que ha sido diseñada con este propósito ubicado en la plataforma GitHub<sup>21</sup>, el nombre que se le ha dado es el de SoftModem. Esta librería no es una librería oficial de Arduino, sino que desarrolladores como ARMS22, propietarios del hilo encargado del desarrollo de esta librería, la han ido creando y modificando.

En la librería se pueden modificar varios datos fundamentales como son las frecuencias a las que se quiere transmitir, o los pines para hacerlo. Para adaptar el *pinaje* a nuestro diseño se deben cambiar las definiciones correspondientes en el archivo SoftModem.cpp. Para cambiar las frecuencias de la modulación FSK e debe acudir al archivo SoftModem.h. Hay que hacer el cambio correspondiente en la parte de código del teléfono.

### 3.2.2.2. Aplicación Android.

Una vez explorados los dos apartados anteriores se va a proceder a explicar el funcionamiento para la decodificación de los datos provenientes de la estación meteorológica. Conviene recordar que hay librerías que se han aprovechado de una aplicación ya existente de un Proyecto de Fin de Carrera<sup>22</sup> al que se le han hecho modificaciones para adaptarlo a las necesidades que se tienen en este trabajo.

A continuación se muestra una figura (Ilustración 51) con la interfaz de usuario que muestra la aplicación para tener una visión de cómo va a trabajar la aplicación.

<sup>21</sup> <https://github.com/arms22/SoftModem> Última vez consultada: 14/06/16

<sup>22</sup> <http://androino.blogspot.com.es/p/project-description.html> Última vez consultada: 15/05/2016



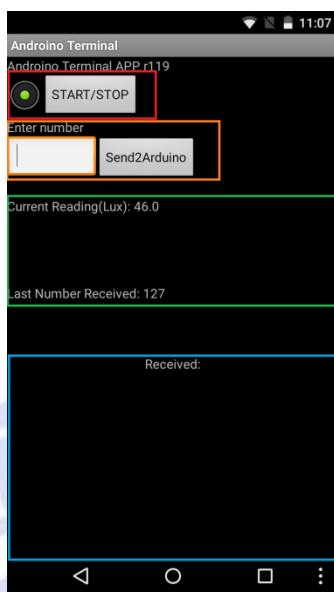


Ilustración 51. Interfaz de usuario AndroidTerm adaptada

En la imagen anterior se pueden distinguir 4 partes:

- **Cuadro Rojo:** botón que habilita la recepción o deshabilita la recepción de datos al teléfono, la interfaz final este botón será eliminado, ya que se ha implementado funciones para que habilite y deshabilite el micrófono de manera automática.
- **Cuadro Naranja:** Es un cuadro de texto que permite el envío de datos desde el teléfono.
- **Cuadro Verde:** En esta zona será donde se visualicen los últimos datos de las mediciones.
- **Cuadro Azul:** Esta zona está destinada para la depuración de fallos en la aplicación. Se mostrarán en esa zona, si el mensaje recibido ha sido correcto, y si no lo es, las razones por las que no se ha podido recibir bien. En la interfaz final también será eliminado.

El código utilizado para la decodificación es bastante complejo por lo que se va a tratar de explicar paso a paso.

Toda la comunicación se basa en la recepción de números enteros. Cada sensor está identificado con un número entero, que se envía antes de los datos.

En el momento en que la aplicación es encendida, habilita el micrófono, y permanece a la espera de notar cambios en la señal recibida en el micrófono. Cuando se recibe un mensaje se inicia el protocolo de extracción del mensaje.

- Se guarda el sonido recibido por el micrófono.
- Se convierte en una tabla de datos.
- Dentro de la tabla de datos se detectan los picos de la onda.
- En función de donde estuvieran ubicado los picos se convierte en una cadena de bits.
- Dentro de la cadena de bits, se extrae el mensaje, detectando el bit de inicio y el de parada.
- Se envía el mensaje obtenido a la *activity* principal.

### 3.3. Otras funciones incorporadas.

#### 3.3.1. Almacenamiento de datos.

Como ya se ha mencionado anteriormente la finalidad del proyecto es adquirir datos para analizar si las variables ambientales afectan a las migrañas. Para este propósito es necesario, que todos los datos que sean recibidos se almacenen de alguna manera. En este proyecto se ha optado por almacenarlos según se recibe una tanda de datos en un fichero en la memoria interna del teléfono.

En el momento en que finaliza la comunicación, todos los datos son introducidos en una nueva línea de un archivo CVS (Comma Separated Value). El orden con el que se graba es, primero, los datos de tiempo, en formato fecha y hora del momento en que se han captado los datos. A esta información le siguen las variables de temperatura, humedad, presión, luz, y ruido, en ese orden.

Por último para que los datos estén disponibles por el equipo de trabajo del grupo *GreenDISC*, el archivo generado por la aplicación, se sube por FTP (File Transfer Protocol) al servidor. La información del servidor es actualizada también cada vez que llega el código de final de comunicación.



## Capítulo 4. Implementación final y pruebas de funcionamiento.

Hasta este punto ya se ha explicado todas las implementaciones que se han realizado en el proyecto, y ahora que se conocen los detalles se realizará una recapitulación para poder tener una visión general del proyecto

### 4.1. Implementación final del sistema electrónico.

Para poder visualizar el sistema hacemos referencia al diagrama presentado en el apartado 2.1.

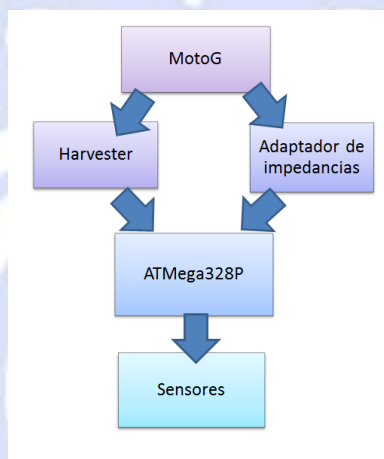


Ilustración 52 Diagrama proyecto

Conectado al teléfono se tiene el *harvester*, que se ha podido ver que como potencia máxima 9,74 mW, que equivalen a unos 2.01 mA, aunque al haber la tensión de alimentación a 3,3V, la potencia entregada por el *harvester* cae a 8.59 mW, se entrega una tensión de alimentación de 3.21V y una corriente de 2.6 mA. Es una corriente de alimentación insuficiente para alimentar el microcontrolador que, como también se ha visto, consume por sí solo 3 mA. La alimentación a través del *harvester* queda descartada, y de manera temporal se ha optado por la alimentación a través de una pila.

Con respecto al adaptador de impedancias y la comunicación FSK tras los cambios realizados, la comunicación de los datos de los sensores se realiza correctamente, finalmente la comunicación será unidireccional. Quedando diseñado la comunicación en sentido contrario por si en un futuro se necesitara. Los datos los envía el microcontrolador y los recibe el teléfono.

El microcontrolador posee el listado de pines de acuerdo a la tabla del [Anexo 6](#).

Finalmente está la placa resultante de todo el diseño se muestra en la Ilustración 53.

Conector *jack*

Adaptador de impedancias

Sensor de Humedad y Temperatura

Sensor de presión

Sensor de nivel de presión sonora

Circuito *harvester*

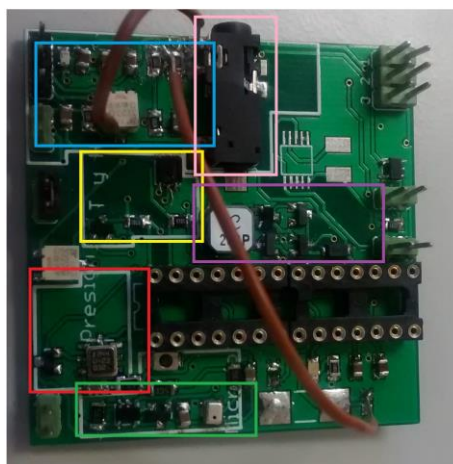


Ilustración 53. Fotografía placa de pruebas

Y el sistema completo queda de la siguiente después del diseño de una carcasa con una impresora 3D, lo cual puede verse en el [Anexo 8](#).

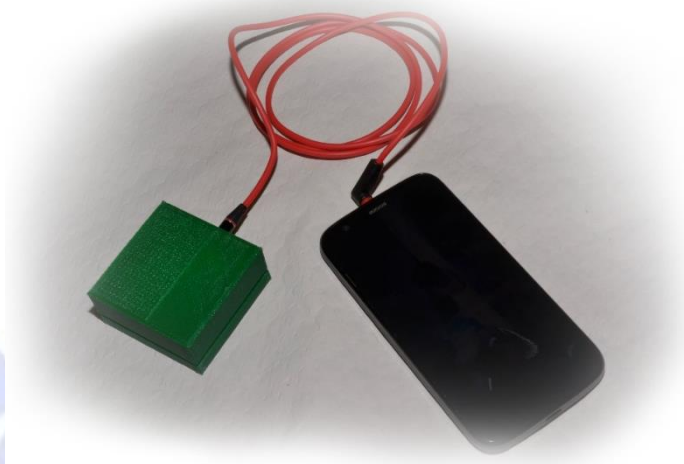


Ilustración 54. Sistema completo

## 4.2. Implementación final del sistema software.

De cara a la interfaz de usuario se le han terminado de modificar ciertas funcionalidades, principalmente se ha eliminado la parte de la interfaz donde se imprimían códigos de *debug*. También se ha eliminado el botón para enviar datos al microcontrolador, ya que aunque se envía un mensaje al microcontrolador para despertarlo y solicitarle los datos, este dato se manda de manera automática y este botón ya no es necesario.

### 4.2.1. Tiempo de envío.

Inicialmente se tenía programado que cada dato se mandara una vez por segundo. De esta manera, todos los mensajes enviados se recibían, pero el índice de decodificación correcto



estaba alrededor del 20 %. Sin embargo se descubrió que si los mensajes se mandaban cada 0.25 segundos, aunque no todos los mensajes que se mandaban se llegaban a recibir; sólo una pequeña parte se pierden mientras que el ratio de decodificación correcta mejoraba considerablemente (alrededor del 75%). Por tanto se decidió implementar esta segunda opción como comunicación final.

## 4.3 Pruebas de Funcionamiento.

En los siguientes apartados se podrán visualizar las pruebas de test del sistema completo.

### 4.3.1. Pruebas de la electrónica del sistema.

Tras el montaje de la placa de pruebas y alimentándolo independientemente se realizaron las pruebas de consumo. Midiendo con un amperímetro, se logró ver que el consumo del microcontrolador se encuentra en 4.7 mA. Como se puede ver en el video del [Anexo3.6](#). Este es un valor muy alto para la corriente suministrada por el *harvester*. Por lo que se deja en la placa el *harvester* para poder realizar estudios y mejoras futuras que por tiempo se escapan del alcance de este Trabajo Fin de Grado.

Por otro lado también se realizaron pruebas a los sensores de la placa.

En cuanto al funcionamiento del micrófono se realizó la siguiente prueba de la captación de sonido.



Ilustración 55. Prueba de medición de sonido

Utilizando una onda de 1 kHz se establece un sonido constante. Este sonido es captado simultáneamente por nuestro micrófono y mediante una aplicación Android que mide el nivel de ruido, y se contrasta con el obtenido con nuestra aplicación, la cual esta puenteada para que muestre el ruido medido por el puerto serial, usando la placa Arduino como adaptador serie-USB. Este valor es mostrado después de que hayan transcurrido los 10 segundos de captación. Este proceso se repite con otros niveles de sonido, hasta llegar al nivel de 70 dB, pudiendo apreciar que la diferencia de ruido medida es mínima. Esta prueba se puede ver en video en el [Anexo 3.3](#).

Por último, como ya se ha visto anteriormente se ha realizado las pruebas de funcionamiento del sensor BMP180 en el [Anexo 3.2](#)

## 4.3.2. Pruebas experimentales.

### 4.3.2.1. Pruebas de comunicación.

Para comprobar el correcto funcionamiento de la comunicación del sistema se realizaron dos pruebas. Primeramente se evaluó la comunicación haciendo que se enviaran los mensajes desde el microcontrolador al teléfono y además el microcontrolador mostrara por el puerto serie qué mensaje había mandado, para comprobar que el dato recibido fuera correcto, esto se realizó utilizando la placa Arduino como adaptador USB - Serie.

Una vez se comprobó el correcto funcionamiento de esta prueba se quitó la placa Arduino y se dejó funcionando el sistema por sí mismo con la alimentación de la batería. Esta prueba también resultó satisfactoria.

En los [Anexos 3.4](#) y [3.5](#) se pueden consultar los videos de estas pruebas.

### 4.3.3. Situación real.

También se han realizado mediciones del sistema probándolo en lo que sería una situación común. Esta situación simula el momento de tomar una ducha. Con esta prueba se evalúa el funcionamiento del sistema.

En esta prueba lo que se ha realizado es aumentar la humedad relativa y la temperatura de una sala cerrada, en un determinado momento se abre la sala. De manera adicional, para comprobar también los sensores de luz y ruido, la prueba comienza con la estancia a oscuras y con un ruido de agua cayendo y luego se corta el ruido y se enciende la luz. Los resultados se muestran en la Ilustración 56.

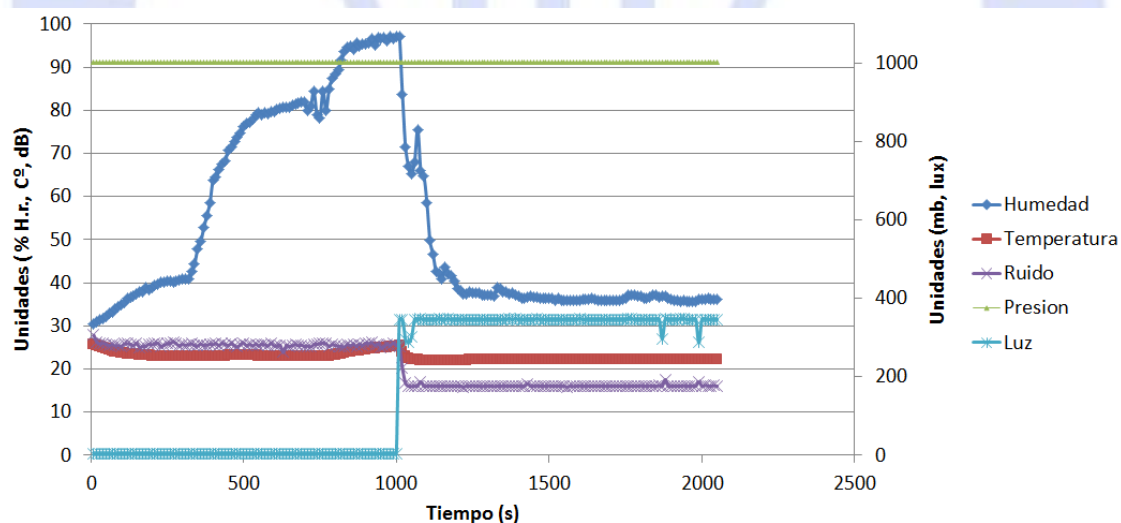


Ilustración 56. Comparativa de las 5 variables

## Capítulo 5. Conclusiones y posibles líneas futuras.

### 5.1. Conclusiones.

El objetivo principal planteado al inicio del diseño e implementación de la estación meteorológica se ha podido resolver satisfactoriamente.

De los objetivos marcados como secundarios, el que se centraba en la captación de los datos y la comunicación con el teléfono también se ha resuelto correctamente. Otro objetivo que se ha podido resolver es el del almacenamiento.

Por otro lado, no se ha podido dar la solución esperada al problema de la alimentación. Se ha reducido el consumo total de la placa pero no se ha podido reducir en la medida necesaria. La energía generada con el *harvester* es insuficiente para alimentar el microcontrolador, aunque se le ha dado una solución temporal incorporándole la alimentación a través de la batería ayudando a contribuir con el objetivo de que sea un sistema portable.

Con respecto al tamaño se ha cumplido con el objetivo, e incluso si se precisara es posible compactar un poco más el tamaño de la placa, por ejemplo sustituyendo el microcontrolador por uno SMD.

A lo largo de este proyecto han surgido problemas debido a la utilización del micrófono del teléfono: mientras el teléfono esté cargando, se incrementa mucho los fallos de comunicación, ya que en el micrófono se introduce una tensión adicional de offset que no se ha logrado averiguar porque ni si ocurre en otros terminales. A su vez, se limitan algunas funcionalidades del usuario, por ejemplo no se pueden grabar videos o realizar llamadas mientras la estación esté en funcionamiento, ya que el micrófono está en uso.

Hay que decir que no se han alcanzado los objetivos que nos planteamos desde un inicio porque el sistema es más complejo de lo esperado y se han dejado cosas para que se hagan más adelante (líneas futuras, ver apartado 5.2).

En el plano personal ha sido un proyecto muy positivo ya que he podido aplicar algunos conocimientos adquiridos durante la carrera, como puede ser Electrónica Analógica, o Microprocesadores. Además, también valoró positivamente la evolución tenida a lo largo del proyecto, ya que ha habido ramas de las que prácticamente se ha tenido que empezar de cero, como en el tema de Android.

### 5.2. Líneas Futuras.

De este proyecto surgen varias líneas por las que se puede seguir trabajando:

- Por un lado se puede trabajar en la captación de energía del *harvester* y tratar de incrementarla para que el microcontrolador y los sensores lleguen a estar alimentados por el teléfono móvil.
- Por otro lado sería necesario trabajar en la reducción de la frecuencia de funcionamiento del microcontrolador, ya que existe mucho margen para reducir el consumo y llegar a estar en niveles que el *harvester* pueda llegar a alimentar.
- Además se puede trabajar para terminar de solucionar que el teléfono pueda realizar el intercambio entre habilitar y deshabilitar el micrófono para la comunicación de forma automática.

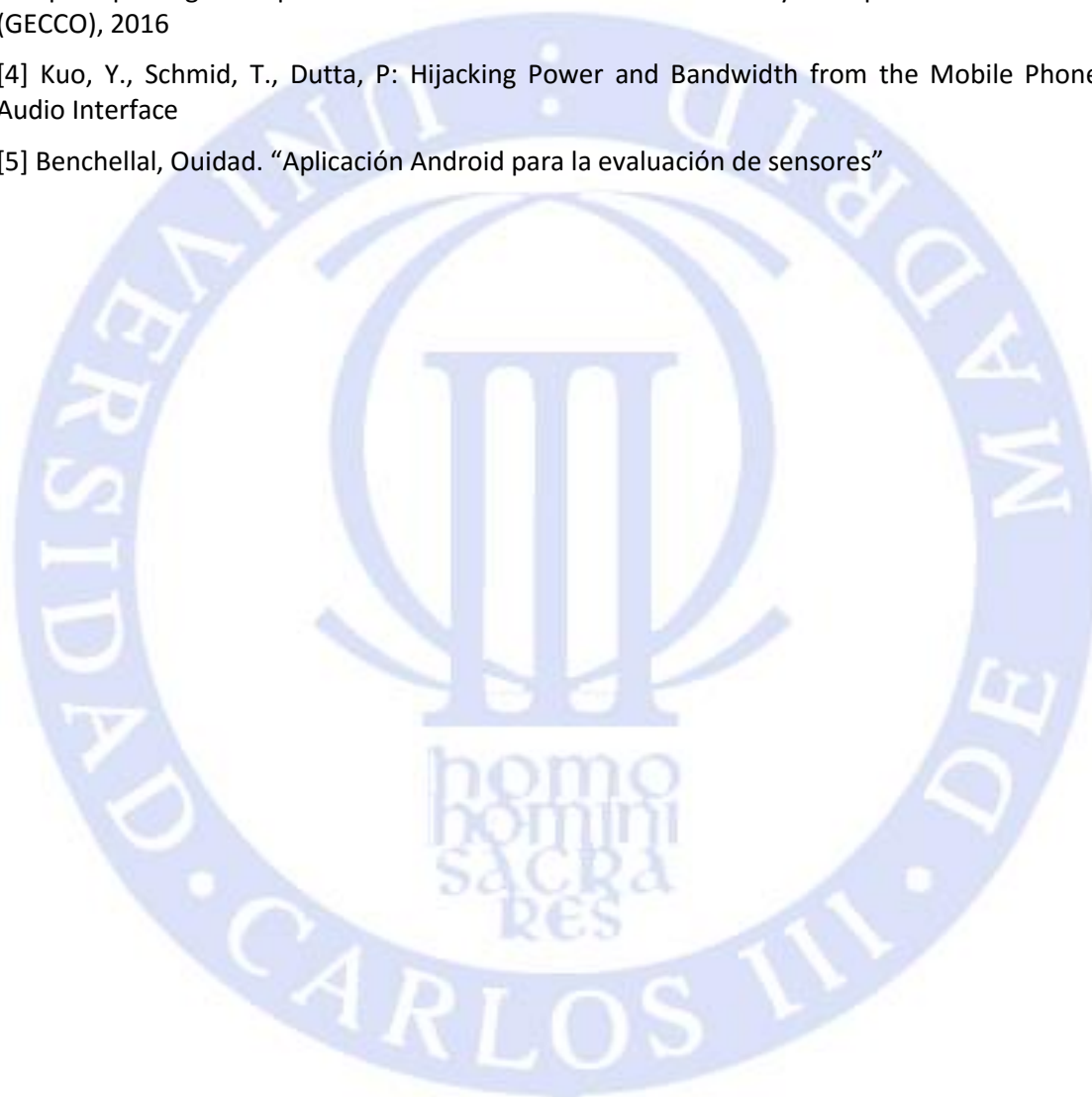
- Continuando con la aplicación Android se debe terminar de integrar la aplicación de captación de datos en la aplicación *“Migraine”* que actualmente usa el equipo de investigación.
- Por último, para hacer más cómodo el dispositivo de cara al usuario se podría poner un sistema para que encienda y apague la estación meteorológica de forma manual para que de esta manera se puedan utilizar las demás funciones que se requieran del teléfono de forma puntual cuando el usuario lo requiera.





## Bibliografía.

- [1] Lipton, R.B.; Scher, A.I. Epidemiology and Economic Impact of Migraine. Curr. Med. Res. Opin. 2001, 17, s4–s12.
- [2] Pagán, J., de Orbe, M. Irene, Gago, A., Sobrado, M., Vivancos Mora, J.A., Risco-Martín, J.L., Moya, J.M., Ayala, J.L.: Robust and Accurate Modeling Approaches for Migraine Per-Patient Prediction from Ambulatory Data, Sensors 2015, 15(7), 15419-15442
- [3] Pagán, J., Risco-Martín, J.L., Moya, J.M., Ayala, J.L.: Grammatical evolutionary techniques for prompt migraine prediction. In: Genetic and Evolutionary Computation Conference (GECCO), 2016
- [4] Kuo, Y., Schmid, T., Dutta, P: Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface
- [5] Benchellal, Ouidad. "Aplicación Android para la evaluación de sensores"





## Listado de acrónimos.



<b>ACK</b>	Acknowledge
<b>ADC</b>	<i>Analog-to-Digital Conversion</i>
<b>dB</b>	Decibelio
<b>DIY</b>	<i>Do It Yourself</i>
<b>EAGLE</b>	Easily Applicable Graphical Layout Editor
<b>FSK</b>	Frequency Shift Keying
<b>GDAF</b>	Grupo de Displays y Aplicaciones Fotónicas
<b>I<sup>2</sup>C</b>	Inter-Integrated Circuit
<b>IDE</b>	Integrated development environment
<b>ISP</b>	In-System Programming
<b>PCB</b>	Printed Circuit Board
<b>R.H.</b>	Relative humidity
<b>SCL</b>	Serial Clock
<b>SDA</b>	Serial Data
<b>SMD</b>	Surface Mounted Device
<b>SO</b>	Sistema Operativo
<b>SPI</b>	Serial Peripheral Interface
<b>SPL</b>	Sound Pressure Level
<b>SS</b>	Slave select
<b>USB</b>	Universal Serial Bus
<b>WSN</b>	Wireless Sensor Network







# Anexos

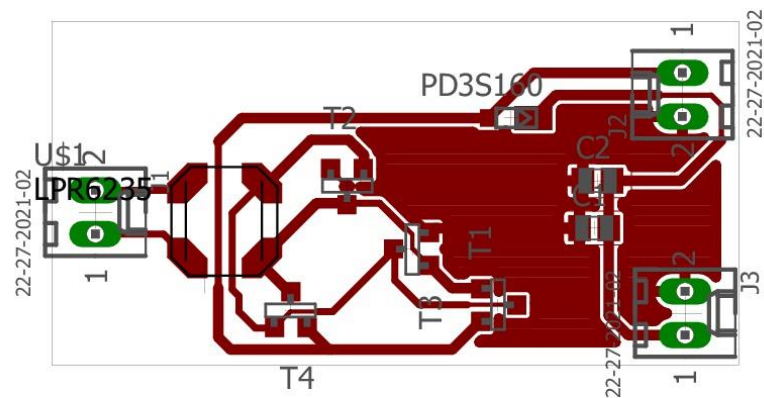


### A.1.1. Esquemático estación meteorológica.



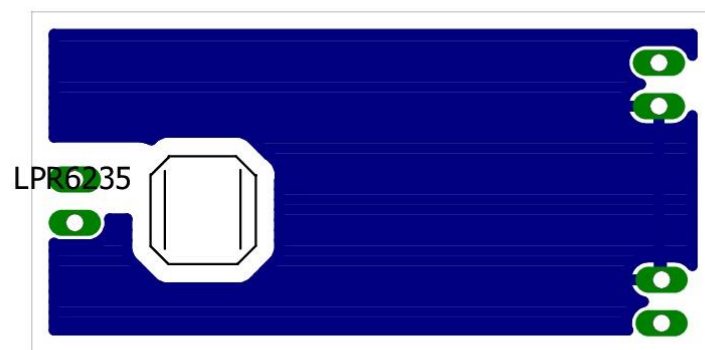


### A.1.2. *Layout circuito harvester capa TOP.*





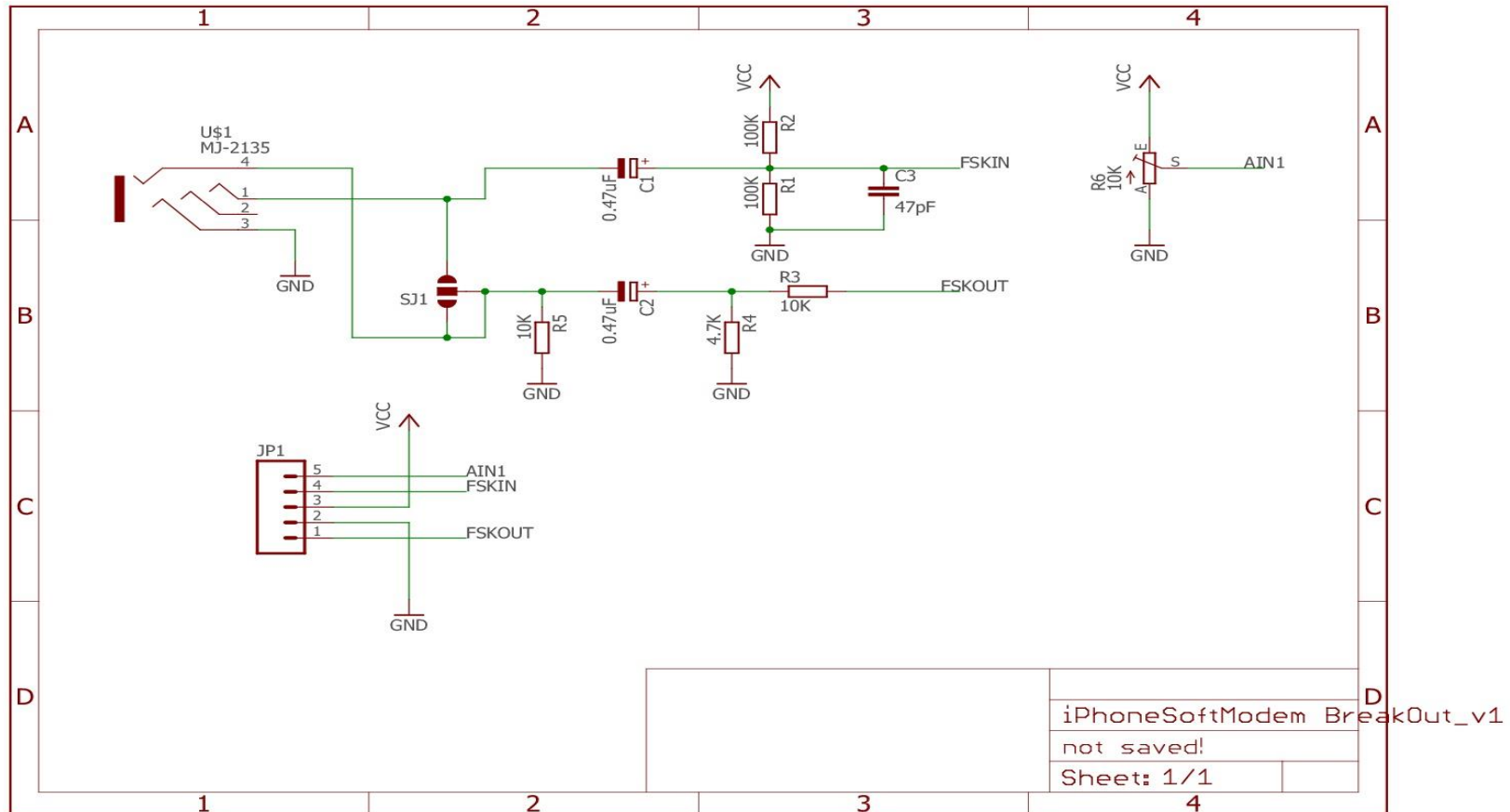
### A.1.3. Layout circuito *harvester* capa BOTTOM.





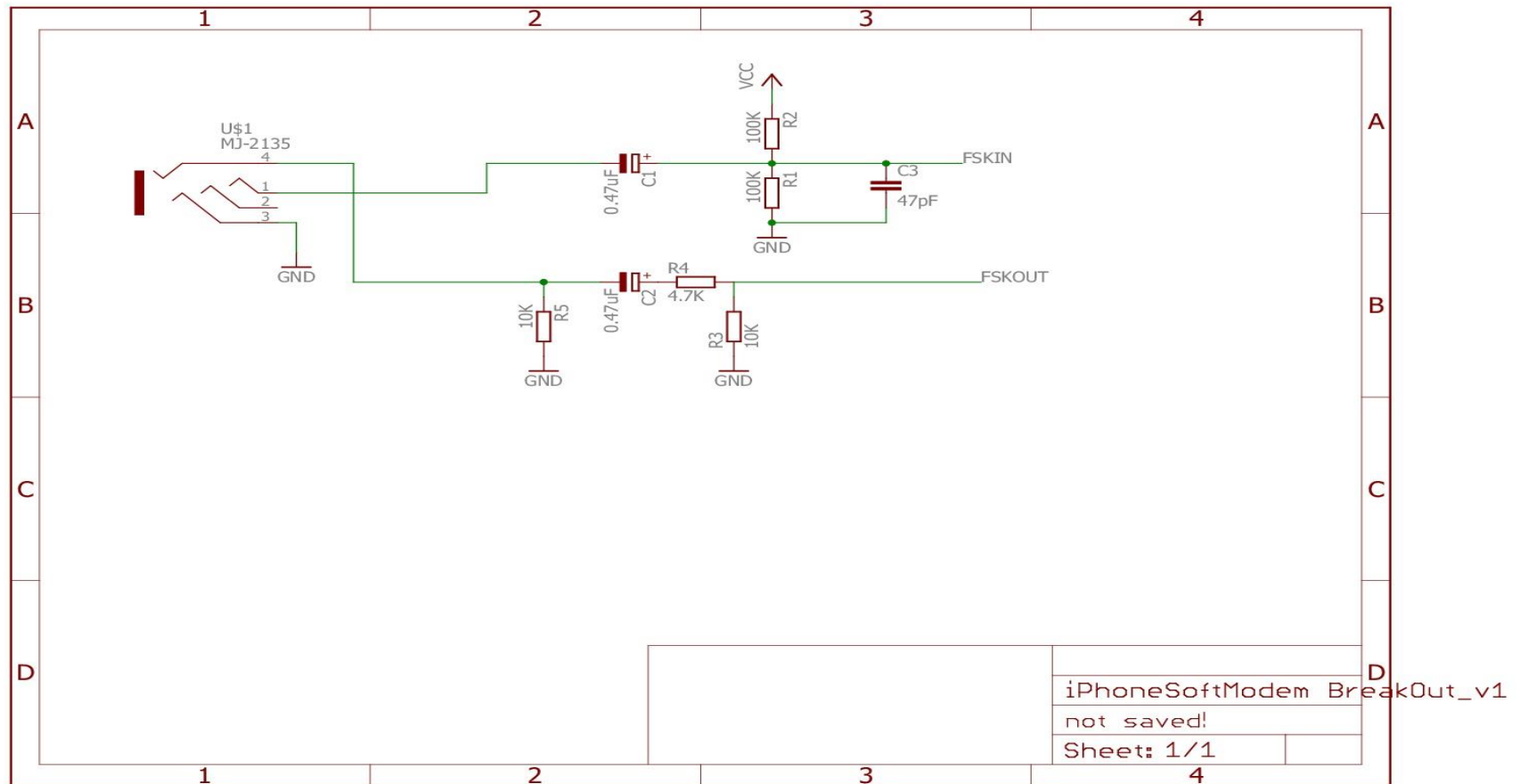


#### A.1.4. Esquemático adaptador de impedancias original.





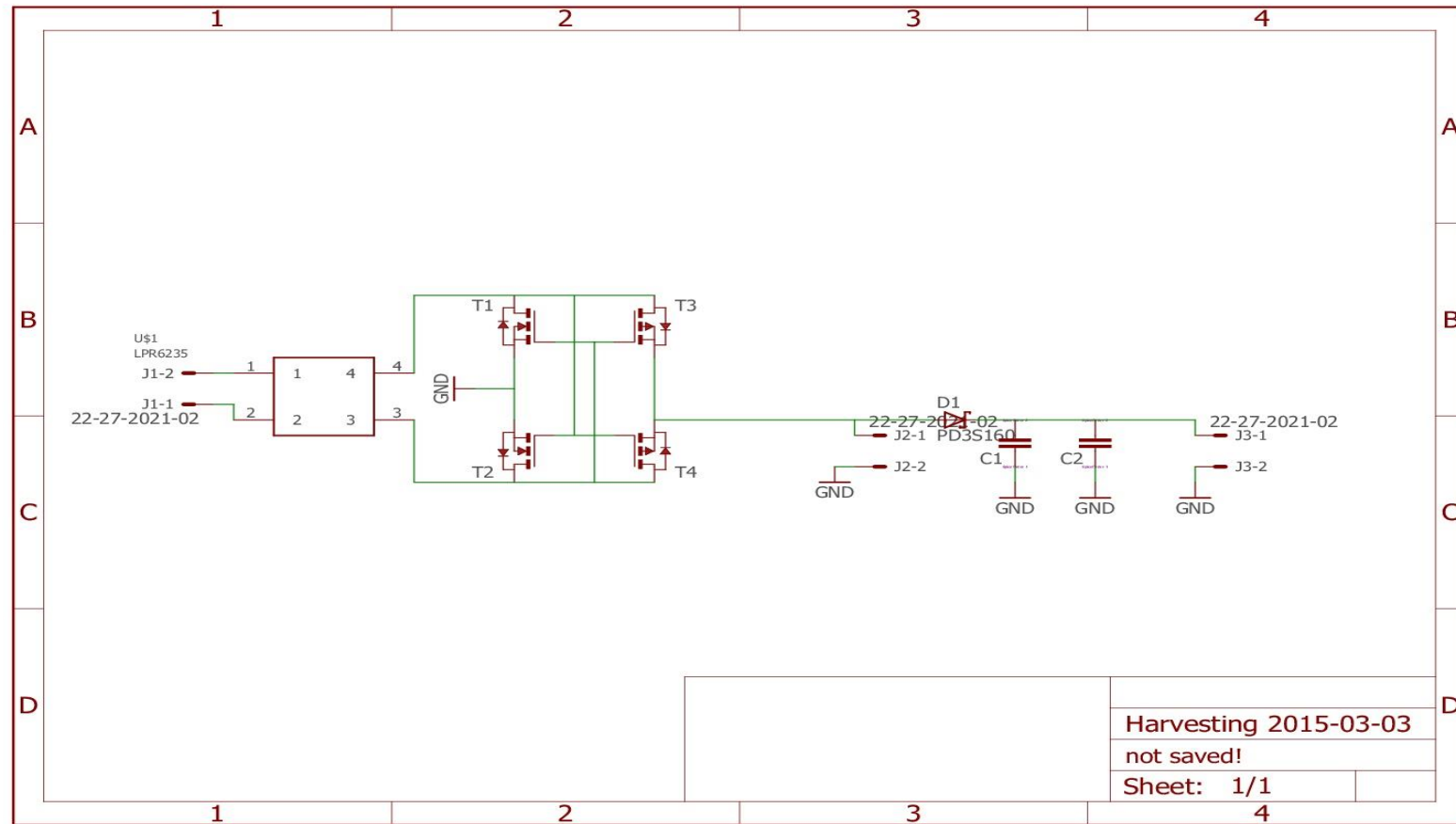
### A.1.5. Esquemático adaptador de impedancias adaptado.





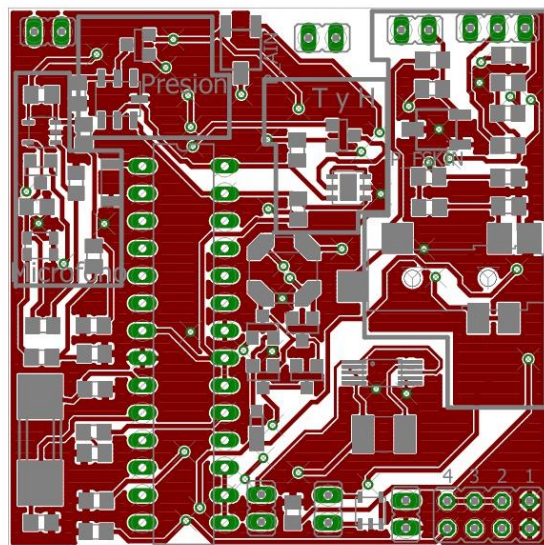


### A.1.6. Esquemático circuito harvester.





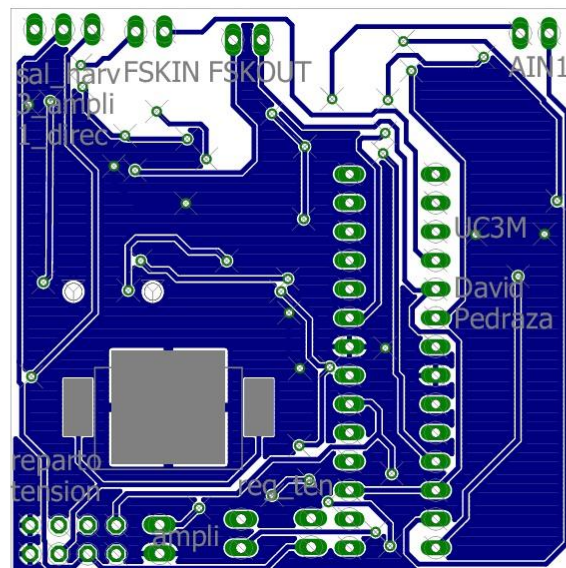
### A.1.7. Layout estación meteorológica capa TOP.







### A.1.8. Layout estación meteorológica capa BOTTOM.





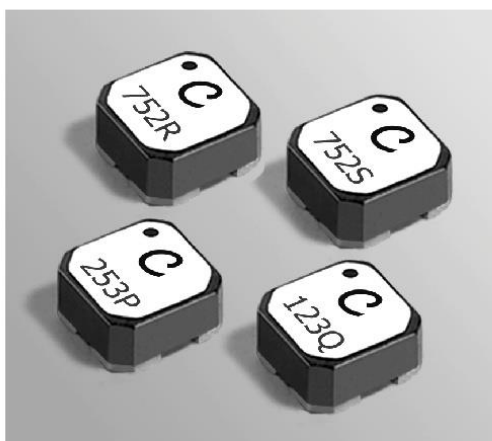
## A.2 Enlaces a hojas de características.

### A.2.1.Datasheet LPR6235.

Document 752R-1



# Coupled Inductors – LPR6235 For Step-Up, Resonant & Flyback Applications



- Can be used as step-up or flyback transformers in DC-DC converters or as autotransformers
- Perfect for low voltage step-up in energy harvesting applications
- Selected by Linear Technology for the LTC3108 and LTC3109 Ultralow Voltage Step-Up Converter and Power Managers

These shielded parts are only 3.5 mm high and 6 mm square. The excellent coupling coefficient ( $k = 0.95$ ) makes them ideal for use in a variety of applications. They can be used as flyback transformers and step-up pulse transformers.

The high Isat and low DCR ratings of these parts provide high efficiency and excellent current handling in a rugged, low cost design.

Custom inductance values and turn ratios may be available upon request.

Part number <sup>1</sup>	Turns ratio	Primary (L1) inductance <sup>2</sup> ±20% (μH)	DCR max (Ohms)		SRF typ <sup>3</sup> (kHz)	Isat <sup>4</sup> (A)
			L1	L2		
LPR6235-253LMR_ 1 : 10		25	0.74	13.7	1300	1.3
LPR6235-253PMR_ 1 : 20		25	0.20	72	580	0.7
LPR6235-123QMR_ 1 : 50		12.5	0.085	200	360	0.9
LPR6235-752RMR_ 1 : 90		7.5	0.085	285	257	1.6
LPR6235-752SMR_ 1 : 100		7.5	0.085	340	230	1.6

1. When ordering, please specify **packaging code**:

**LPR6235-253PMRC**

**Packaging:** C = 7" machine-ready reel. EIA-481 embossed plastic tape (350 parts per full reel).

B = Less than full reel. In tape, but not machine ready. To have a leader and trailer added (\$25 charge), use code letter D instead.

D = 13" machine-ready reel. EIA-481 embossed plastic tape. Factory order only, not stocked (1500 parts per full reel).

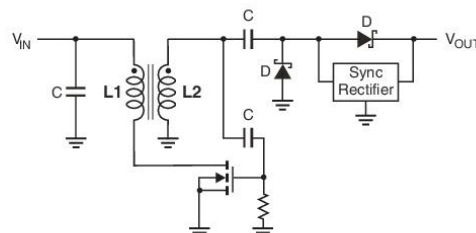
2. Inductance is measured at 100 kHz, 0.1 Vrms, 0 Adc on an Agilent/HP 4284A LCR meter or equivalent.

3. SRF measured using an Agilent/HP 4191A or equivalent.

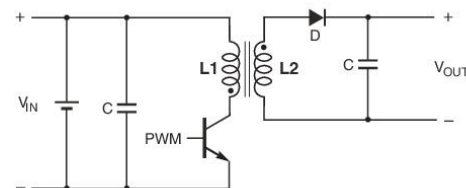
4. DC current applied to L1, at which the inductance drops 10% from its value without current.

5. Electrical specifications at 25°C.

Refer to Doc 362 "Soldering Surface Mount Components" before soldering.



Typical Step-Up Converter



Typical Flyback Converter



US +1-847-639-6400 sales@coilcraft.com  
 UK +44-1236-730595 sales@coilcraft-europe.com  
 Taiwan +886-2-2264 3646 sales@coilcraft.com.tw  
 China +86-21-6218 8074 sales@coilcraft.com.cn  
 Singapore +65-6484 8412 sales@coilcraft.com.sg

Document 752R-1 Revised 08/06/14

© Coilcraft Inc. 2016  
 This product may not be used in medical or high risk applications without prior Coilcraft approval. Specification subject to change without notice. Please check web site for latest information.





## **A.3. Videos de pruebas.**

### **A.3.1.Video alimentación placa de pruebas.**

<https://www.youtube.com/watch?v=Lv8piWyqmck>

En este video se muestra como, la placa de pruebas implementada, es capaz, de con una alimentación mediante una pila, alimentar el microcontrolador ATmega328 para que este haga parpadear un LED.

### **A.3.2.Video prueba sensor BMP180.**

[https://www.youtube.com/watch?v=F1A\\_4Zra83Y](https://www.youtube.com/watch?v=F1A_4Zra83Y)

En este video se muestra como, alimentando la placa de pruebas desde la toma de 3.3V de la placa de Arduino, se requiere al sensor de presión BMP180 instalado en la placa de pruebas los datos sobre la temperatura y la presión atmosférica. Estos datos son recibidos por el microcontrolador.

Una vez recibidos, el microcontrolador, utilizando el puerto serie, y la placa Arduino como Adaptador serie-USB, muestra por pantalla los datos de las variables.

### **A.3.3.Video calibración sensor de nivel de presión sonora.**

<https://www.youtube.com/watch?v=6AWGmKEZthI>

En este video se muestra como se utiliza el sensor de nivel de presión sonora, el cual se conecta con el microcontrolador ubicado en una placa protoboard. Para poder calibrarlo se compara con el nivel de ruido medido con una aplicación instalada en el teléfono. Esta prueba se repite dos veces, una con una intensidad sonora de 40 dB y otra de 70 dB.

### **A.3.4.Video prueba de comunicación con monitor serial.**

<https://www.youtube.com/watch?v=-lO6vR-226g>

En este video se muestra la comprobación realizada de cara a la comunicación el microcontrolador al mismo tiempo que manda mensajes al teléfono móvil, muestra por la pantalla del ordenador el número que se manda, a través de la placa Arduino, que actúa como adaptador serie-USB.

De esta manera el mensaje que se visualiza en el móvil que comprueba que corresponde con el mostrado en la pantalla.

### **A.3.5.Video prueba de comunicación del sistema completo.**

<https://www.youtube.com/watch?v=iHqRxURDAso>

Una vez realizada la comprobación del apartado anterior, se comprueba, que el sistema funciona correctamente solo. Este video es la prueba de ello.

### **A.3.6.Video prueba de consumo microcontrolador.**

[https://www.youtube.com/watch?v=wGgV\\_PFDd2k](https://www.youtube.com/watch?v=wGgV_PFDd2k)

En este video se muestra la prueba realizada con un amperímetro, para medir el consumo del microcontrolador.

### **A.3.7.Video prueba dormir microcontrolador.**

[https://www.youtube.com/watch?v=ogy1x5\\_agLQ](https://www.youtube.com/watch?v=ogy1x5_agLQ)

En este video se muestra como midiendo el consumo del microcontrolador, se demuestra cómo se duerme el microcontrolador. Éste se vuelve a despertar y hace parpadear un led cuando detecta una señal en su pin que en este caso proviene de un botón.

### **A.3.8.Video prueba despertar microcontrolador desde el teléfono.**

<https://www.youtube.com/watch?v=CPYtSB75stc>

Partiendo del video del apartado anterior, en este video se observa como desde el teléfono se consigue despertar el microcontrolador para que haga parpadear un led.



## A.4. Otras funcionalidades hardware.

Además de las funciones ya mencionadas en la memoria, tratando de ahorrar en el consumo de la estación meteorológica, también se han desarrollado otras funciones que se detallan a continuación.

Para cada sensor se ha incorporado un transistor de corte para evitar que cuando no estén realizando medidas continúen consumiendo energía. El tipo de transistor que se ha seleccionado es un MOSFET tipo P.

Además, buscando también el ahorro del consumo del sistema se ha desarrollado otras funcionalidades, como puede ser la reducción de frecuencia en el microcontrolador, o dormir el propio microcontrolador.

### A.4.1. Reducción de frecuencia ATmega328P.

Por otro lado, como se ha mencionado anteriormente, también se ha tratado de reducir el consumo por parte del microcontrolador que gestiona la estación meteorológica. En un principio se extrajo el microcontrolador de Arduino para ubicarlo en la estación meteorológica y así ganar en tamaño. Para ello se necesita proveer al microcontrolador de un reloj externo y alimentarlo mediante una pila de 9V junto con un regulador de tensión que fija la tensión en 5V (Ilustración 57).

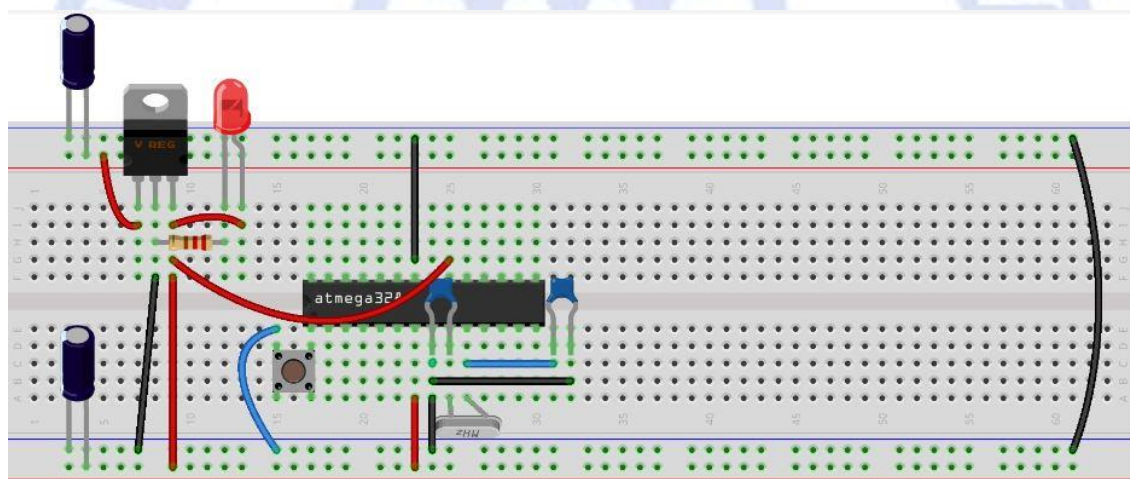


Ilustración 57. Microcontrolador en Protoboard

A continuación, para disminuir la frecuencia de funcionamiento del microcontrolador se modifica el *bootloader* que el IDE de Arduino carga a nuestro ATmega328 y que no es necesario<sup>23</sup>

Primero se debe cargar el código llamado ArduinoISP ubicado en los ejemplos del Arduino IDE. Paralelamente, se debe modificar el archivo en el que se basará Arduino para grabar el *bootloader* en el microcontrolador. La ruta de acceso a este archivo es C:\Program Files (x86)\Arduino\ hardware\arduino\bootloaders\breadboard\board.txt. y el contenido es el siguiente.

<sup>23</sup> <http://www.cientificosaficionados.com/foros/viewtopic.php?f=7&t=18124>  
visitada:16/06/16

```
atmega328bb.name=ATmega328 on a breadboard (8 MHz internal clock)
```

```
atmega328bb.upload.protocol=stk500  
atmega328bb.upload.maximum_size=30720  
atmega328bb.upload.speed=57600
```

```
atmega328bb.bootloader.low_fuses=0xE2  
atmega328bb.bootloader.high_fuses=0xDA  
atmega328bb.bootloader.extended_fuses=0x05
```

```
atmega328bb.bootloader.path=arduino:atmega  
atmega328bb.bootloader.file=ATmegaBOOT_168_atmega328_pro_8MHz.hex  
atmega328bb.bootloader.unlock_bits=0x3F  
atmega328bb.bootloader.lock_bits=0x0F
```

```
atmega328bb.build.mcu=atmega328p  
atmega328bb.build.f_cpu=8000000L  
atmega328bb.build.core=arduino:arduino  
atmega328bb.build.variant=arduino:standard
```

Ilustración 58. Archivo Bootloader

- **Cuadro Negro:** Es el nombre con el que este nuevo *bootloader* se visualizará en el IDE de Arduino.
- **Cuadro Rojo:** Son las sentencias hexadecimales con las que se configuran los bits del microcontrolador.
- **Cuadro Verde:** Hace referencia al archivo hexadecimal en el que se basará para cargar en el microcontrolador.
- **Cuadro Azul:** Se comunica la frecuencia del reloj a la que funcionará el microcontrolador.

Una vez que se ha editado el fichero, y se ha cargado el código, se debe seleccionar el *bootloader* modificado como tarjeta, y seleccionar el Arduino como ISP en la pestaña de programador realizar el siguiente conexionado.

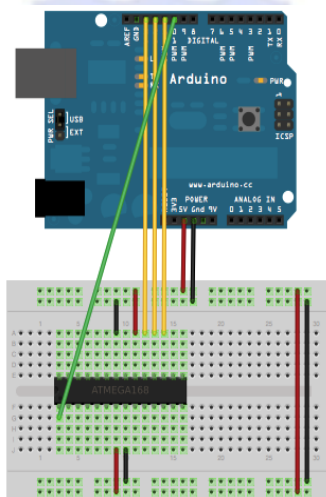


Ilustración 59. Conexionado para cargar el Bootloader

Una vez grabado se debe retirar el microcontrolador de la placa de Arduino y conectar los pines de TX y RX de la placa con los respectivos del microcontrolador. Y cargar el *sketch* que se desee en el microcontrolador como si se estuviera haciendo en la placa.



Este método tiene un inconveniente, solamente sirve para grabar el primer *sketch*, si se quiere sobrescribir el programa, se debe repetir todo el proceso. Con esta medida junto a que la alimentación se ha bajado desde los 5 voltios que tenía la placa Arduino a 3.3 voltios se ha podido reducir el consumo de algo menos de 10 mA a algo más de 3 mA, como se puede ver en la siguiente imagen.

Figure 35-2. ATmega328P: Active Supply Current vs. Frequency (1MHz - 20MHz)

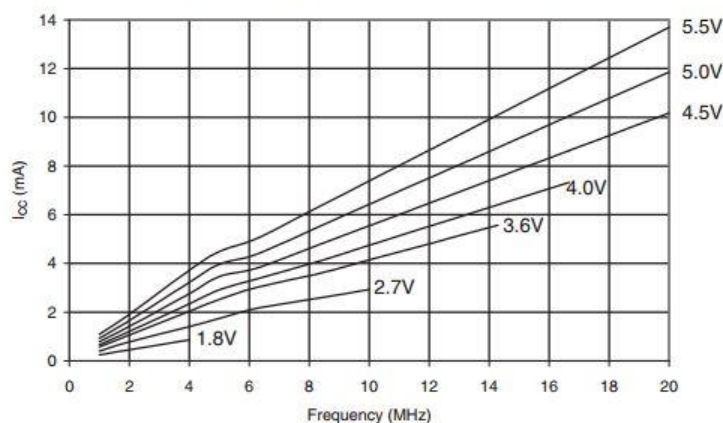


Ilustración 60. Gráfica consumo ATmega328P<sup>24</sup>

Una vez que se tenía grabado el *bootloader* con 8MHz se ha tratado de grabar el *bootloader* con 1 MHz y con 32 kHz, para ello se cambió el archivo anteriormente comentado, observando cuales son los bits referente a la frecuencia del reloj. En el caso de los 32 kHz, no se grabó el *bootloader*. En el caso de 1 MHz se cargó correctamente el *bootloader* pero se presentaron problemas a la hora de grabar el código, ya que hubo problemas de comunicación entre el programa y el ATmega328, posiblemente el cambio de frecuencia de trabajo en el microcontrolador llevo a un cambio en la tasa de envío/recepción de datos en el puerto serie, que es por donde se graba el código, por lo que es una tasa de envío diferente a la que usa la interfaz Arduino para cargar el código.

Por tanto, la frecuencia de trabajo del microcontrolador finalmente es 8 MHz.

#### A.4.2. Dormir microcontrolador y despertarlo desde el teléfono móvil

Ya que no es necesario estar constantemente midiendo las variables, se ha establecido un tiempo en que no se toman datos, como ya se ha visto en el apartado 3.2.1.

Con la intención de reducir el consumo del microcontrolador durante este tiempo se propuso dormir el microcontrolador basándose un experimento previo<sup>25</sup>. En este caso el componente por el cual se despierta el microcontrolador es un botón. El video de esta prueba está disponible en el [anexo 3.7](#).

Una vez comprobado el correcto funcionamiento se procedió a hacer que sea el teléfono quien despierte al microcontrolador, esto se consiguió realizando una serie de cambios en el tipo de

<sup>24</sup> [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf) Última vez consultada: 15/05/2016

<sup>25</sup> <https://holadimake.wordpress.com/2015/06/23/waking-up-an-arduino-with-input-from-a-sensor/> Última vez revisado: 16/06/16



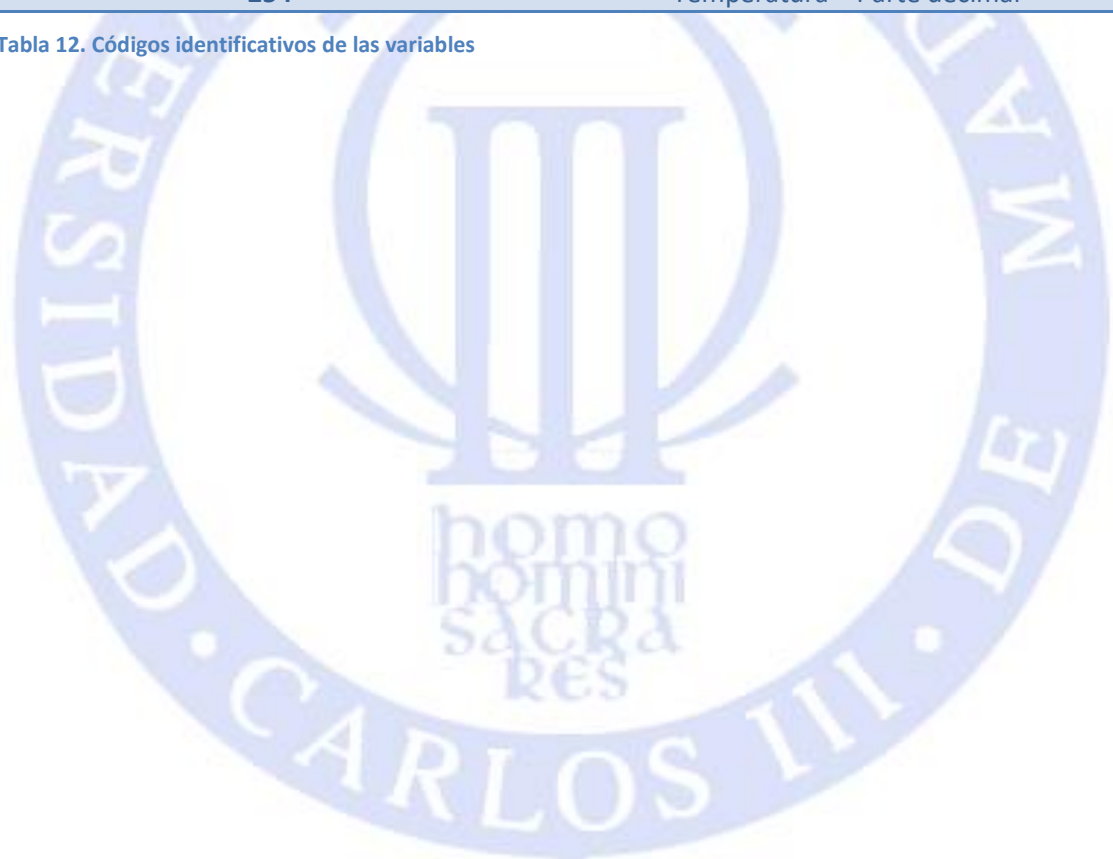
señal que el microcontrolador espera recibir mientras está dormido. El video de esta prueba está disponible en el [anexo 3.8](#).



## A.5. Códigos de identificación de las variables.

Código	Número
246	Fin de envío
247	Ruido – Parte entera
248	Ruido – Parte decimal
249	Presión – Parte entera
250	Presión – Parte decimal
251	Humedad – Parte entera
252	Humedad – Parte decimal
253	Temperatura – Parte entera
254	Temperatura – Parte decimal

Tabla 12. Códigos identificativos de las variables





## A.6. Puertos del microcontrolador.

Pin	Asignación
1	Reset
5	FSKOUT
6	Transistor BMP180
7	3.3V
8	Tierra
11	Transistor SHT21
12	FSKIN
13	AIN1
14	Transistor Micrófono
19	LED
20	3,3V
21	3,3V
22	Tierra
23	ADC
27	I <sup>2</sup> C (SDA)
28	I <sup>2</sup> C (SCL)

Tabla 13. Pines asignados del microcontrolador





## **A.7.Pruebas funcionales que no se pueden incorporar en el diseño.**

Con la intención de informar a las posibles líneas de trabajo que surjan a partir de este proyecto, se quiere incorporar la información acerca de pruebas que se han realizado, que cumpliendo el objetivo principal que persigue la misma, no se han incorporado en el proyecto por afectar a otras facetas de este proyecto.

### **A.7.1. Reducción de frecuencia dinámica ATmega328.**

Como se ha podido ver anteriormente se consiguió reducir la frecuencia de funcionamiento del microcontrolador pero no hasta los valores esperados, por este motivo se probó un método para reducir la frecuencia dinámicamente[35].

Este método consiste en introducir por código, al igual que en el caso del *prescaler* del ADC, instrucciones de precompilador para cambiar la frecuencia de funcionamiento.

Con esta prueba se consiguió el objetivo de reducir la frecuencia del controlador, y con ello se redujo en gran medida el consumo del microcontrolador. Sin embargo esta reducción dinámica de frecuencia tiene unos inconvenientes. Principalmente la reducción de frecuencia conlleva que todo va más lento, todo aquello que dependa de un número concreto de ciclos de reloj teóricamente debe ir en la medida en que se ha reducido la frecuencia del reloj.

Se probó primeramente la comunicación serie del microcontrolador, se redujo la frecuencia de reloj dividiéndola por 2. La comunicación serie se configuró con 9600 baudios. A la hora de comprobar que la comunicación funcionaba, se vio que a 9600 baudios no se recibía información, y si se trata de visualizar el resto del abanico de frecuencias y no se recibe nada en el monitor serial. Esta misma prueba se realizó con la modulación FSK con idéntico resultado. Por lo que se concluye que este método no es posible utilizarlo.



### A.8. Diseño caja de protección.

A esta placa también se le ha decidido crear una carcasa para protegerla de golpes. Para ello se ha utilizado la impresora 3D. Se ha creado en dos piezas por separado, la parte superior y la parte inferior.

Para el diseño de las placas se utilizó el programa SketchUp de diseño en 3D, y las piezas son las siguientes.

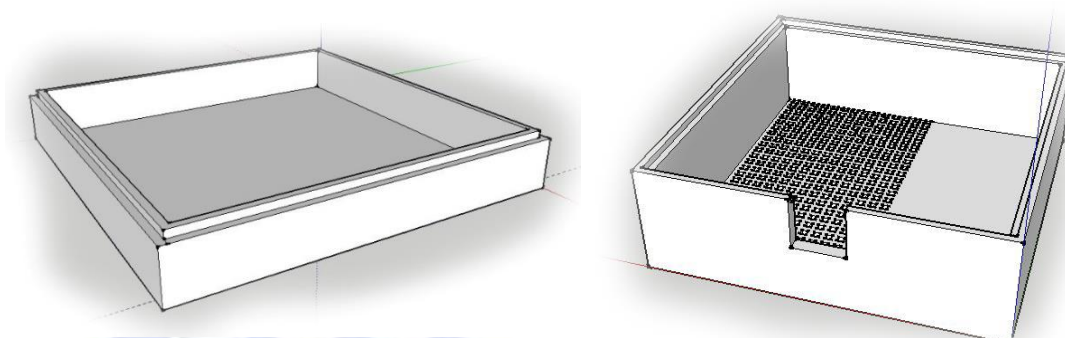


Ilustración 61. Base y tapa de la carcasa

Un detalle importante es la necesidad de ventilación dentro de la carcasa para alterar lo menos posible las medidas de las señales ambientales. Para ello se han realizado orificios en la tapa. El resultado final se puede apreciar en la siguiente fotografía.



Ilustración 62. Fotografía carcasa



## A.9. Pliego de condiciones.

### A.9.1. Pliegos de especificaciones generales.

El ingeniero redactor del Proyecto se reserva el derecho de percibir cuanto ingreso en concepto de derechos de autor pudiera derivarse de una posterior comercialización, reservándose además el derecho de introducir cuantas modificaciones crea conveniente.

### A.9.2. Pliegos de especificaciones técnicas.

En este apartado se exponen las condiciones necesarias e que deben cumplir los componentes.

#### A.9.2.1. Smartphone.

El teléfono deberá presentar las siguientes condiciones mínimas:

- Sistema Operativo Android 4.4.2 *kitkat* o posterior.
- Conector *jack* con habilitación para manos libres.
- Sensor de luz incorporado que cumpla con las condiciones necesarias.

#### A.9.2.2. Estación meteorológica.

##### A.9.2.2.1. Conector *jack*.

El conector debe poseer 4 puertos para poder utilizar la función de micrófono.

##### A.9.2.2.2 Sensores.

Todos los sensores como requisito común, deben ser de bajo consumo.

##### A.9.2.2.2.1. Sensor de humedad.

- Debe ser capaz de medir entre 10% y 90% de R.H.
- Ha de tener una resolución de una décima.

##### A.9.2.2.2.2. Sensor de presión.

- Debe ser capaz de medir entre 700 *mbar* y 1200mb.
- Ha de tener una resolución de 1 *mbar*.

##### A.9.2.2.2.3. Sensor de temperatura.

- Debe ser capaz de medir entre -5 °C Y 45 °C.
- Ha de tener una resolución de una décima de grado centígrado.

##### A.9.2.2.2.4. Sensor de luz.

- Debe ser capaz de medir entre 0 lux y 750 lux
- Ha de tener una resolución de al menos 1 lux.





## A.10. Presupuesto.

Código	Unidades de medida	Descripción	Cantidad	Precio unitario	Importe total
<b>Capítulo 1. Recepción de señal.</b>					
1.01	U	Motorola Moto G con Android v4.3 (Jelly Bean) Pantalla Gorilla Glass 4.5" Procesador Quad Core 1.2Ghz Cámara 5 mpx + Flash LED.	1	189,00 €	189,00 €
<b>Total capítulo 1:</b>					189,00 €
<b>Capítulo 2. Sisma de comunicación.</b>					
2.01	U	Conector <i>Jack</i> TRRS. 3.5mm Macho a Macho 4	1	0,74 €	0,74 €
2.02	U	Pole AUX Coche Conector <i>Jack</i> Audio Cable.	1	2,14 €	2,14 €
2.04	Lote	Cables, resistencias y condensadores etc.	1	3,15 €	3,15 €
<b>Total Capítulo 2:</b>					6,03 €
<b>Capítulo 3. Placas estación meteorológica.</b>					
3.01	U	Microcontrolador ATMega328P	1	3,33 €	3,33 €
3.02	U	Sensor BMP 180	1	3,47 €	3,47 €
3.03	U	Sensor SHT 21	1	5,94 €	5,94 €
3.04	U	Micrófono	1	1,47 €	1,47 €
3.05	U	Amplificadores operacionales TSV621	2	0,85 €	1,70 €
3.06	U	Portapilas	1	1,30 €	1,30 €
3.07	U	Microtransformador	1	1,72 €	1,72 €
3.08	U	Zócalo 28 pines	1	0,80 €	0,80 €
3.09	Lote	Componentes específicos utilizados: Transistores, diodos schotkky etc.	1	4,21 €	4,21 €
3.10	Lote	Componentes básicos: Resistencias, condensadores, headers, etc.	1	3,11 €	3,11 €
3.11	U	Fabricación y envío placas	1	45,00 €	45,00 €
<b>Total Capítulo 3:</b>					72,05 €
<b>Capítulo 4. Materiales para pruebas.</b>					
4.01	U	Placa Arduino Uno Rev 3	1	19,85 €	19,85 €
4.02	U	Placa protoboard	1	6,30 €	6,30 €
<b>Total Capítulo 4:</b>					26,15 €

<b>Capítulo 5. Salario Ingeniero.</b>					
5.01	Horas	Honorarios del ingeniero a cargo del proyecto suponiendo un salario de 2€/hora	585	22,00 €	12.870,00 €
<b>Total Capítulo 5:</b>					<b>12.870,00 €</b>
<b>Importe Total:</b>					<b>13.163,23 €</b>
<b>Total Capítulo 1 - Recepción señal:</b>					<b>189,00 €</b>
<b>Total Capítulo 2 - Sistema de comunicación :</b>					<b>6,03 €</b>
<b>Total Capítulo 3 - Placas estación meteorológica:</b>					<b>72,05 €</b>
<b>Total Capítulo 4 -Material de pruebas:</b>					<b>26,15 €</b>
<b>Total Capítulo 5 - Salario Ingeniero:</b>					<b>12.870,00 €</b>

